(19) Europäisches Patentamt
European Patent Office
Office européen des brevets

(11) EP 0 755 161 A2

(12) EUROPEAN PATENT APPLICATION

(43) Date of publication:
22.01.1997 Bulletin 1997/04

(51) Int Cl.^{6}: H04N 9/79

(21) Application number: 96305213.9

(22) Date of filing: 16.07.1996

(84) Designated Contracting States:
AT DE ES FR GB IT NL

(30) Priority: 18.07.1995 JP 202703/95

(71) Applicant: SONY CORPORATION
Tokyo 141 (JP)

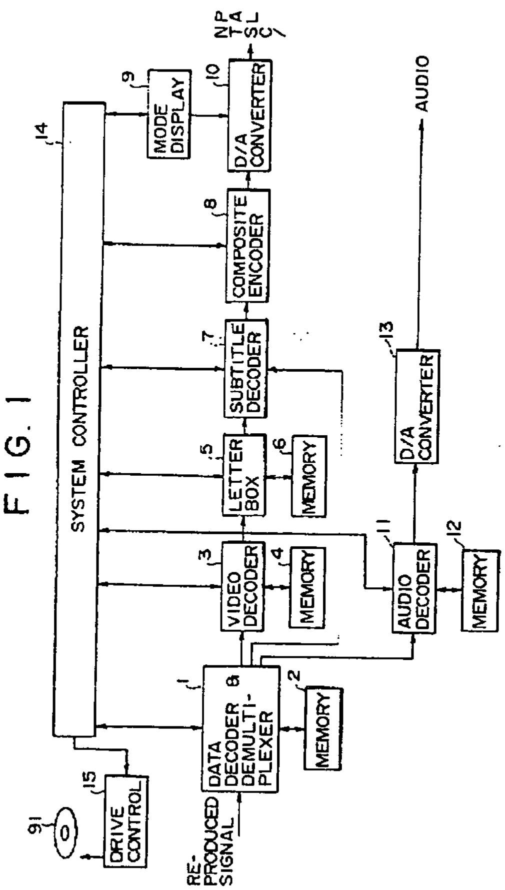
(72) Inventor: Tsukagoshi, Ikuo, c/o Intell. Prop.
Shinagawa-ku, Tokyo 141 (JP)

(74) Representative: Pratt, Richard Wilson et al
D. Young & Co,
21 New Fetter Lane
London EC4A 1DA (GB)

10/825,622

(54) Encoding and decoding data, and data searching

(57) Multiple data stream searching is achieved by encoding a video broadcast into multiple types of data streams including video, audio and subtitle data. Addresses are generated for each of the data streams pointing to a position of the data stream to be stored on a record medium. These addresses are stored in areas of the record medium reserved by type of data stream Such that location of any event represented by data in a data stream can be achieved by locating the data streams at the same type as that of the data to be located. A computer-readable memory directs a computer to store these addresses and to decode data streams at these addresses which correspond to the type of data stream to be searched. The data streams which are of the type to be searched are decoded and the event to be searched is found in these decoded data streams.



F I G. 1

BLOCK DIAGRAM OF INTRA-PLAYER CIRCUITS

EP 0 755 161 A2

Description

The invention relates to a method of an apparatus for encoding data; a method of and apparatus for decoding data, and to searching for encoded data. In embodiments of the invention data is recorded on a record medium and includes subtitle data.

Television broadcasting or video reproduction (such as from a video disk) provides subtitles superimposed on the video image. Problematically, the subtitles are permanently combined with the underlying video image and cannot be manipulated at the receiving (or reproducing) end. The subtitles, for example, cannot be searched for information concerning a specific scene occurring in the video image or sound in its corresponding audio track.

Compact Disc Graphics (CD-G) provide some flexibility in searching subtitles because this technique records graphics in the form of subcodes. However, CD-G has a serious disadvantage because this technique is limited to compact disc (CD) applications, which are slow by television standards. That is, the CD-G technique does not lend itself to manipulation of subtitles in real-time television broadcasts or video reproductions.

As will be shown with reference to Figs. 18A-C and 19, the lead time required to generate a full CD-G screen is grossly inadequate for normal television or video broadcasts. Fig. 18A depicts the CD-G data format in which one frame includes 1 byte of a subcode and 32 bytes of audio channel data, 24 bytes of which are allocated for L and R audio channel data (each channel having 6 samples with 2 bytes per sample) and 8 bytes allocated to an error correction code. The frames are grouped as a block of 98 frames (Frame 0, Frame 1, ..., Frame 96 and Frame 97) as shown in Fig. 18B and eight of these blocks P,Q,R,S,T,U,V and W are transmitted as shown in Fig. 18C. The subcodes for Frames 0 and 1 in each block are reserved for sync patterns S0, S1, whereas the subcodes for the remaining 96 frames are reserved for various subcode data. The first two blocks P, Q are allocated to search data employed for searching through record tracks, while the remaining 6 blocks R,S,T,U,V and W available for graphic data.

CD-G transmits each block of 98 frames at a repeating frequency of 75 Hz. Thus, the data transmission rate for 1 block is (75 Hz x 98 bytes) 7.35 kHz, resulting in a subcode bit rate of 7.35 K bytes/s.

The transmission format for transmitting the information present in blocks R,S,T,U,V and W is shown in Fig. 19, wherein each of the 96 frames (2,3,... 97) of the 6 blocks (R,S,T,U,V and W) are transmitted as a packet including 6 channels (R to W) of 96 symbols per channel (a symbol comprising three cells). The packet is further subdivided into 4 packs of 24 symbols apiece (symbol 0 to symbol 23), with each pack storing a CD-G character. It will be appreciated that, a CD-G character is made up of 6 x 12 pixels and, therefore, is easily accommodated in each 6 x 24 pack. According to the CD-G format, the 6 x 12 CD-G character is stored in the six channels of (R,S,T,U,V and W) at symbols 8 to 19 (12 symbols). The remainder of the symbols in each of the packs store information about the character.

Mode information is one example of information stored in the packs and is stored in the first 3 channels (R, S, T) of symbol 0 in each pack. Item information is another example which is stored in the last 3 channels (U, V, W) of symbol 0. A combination of the mode information and the item information defines the mode for the characters stored in the corresponding pack as follows:

Table 1

Mode	Item	
000	000	0 mode
001	000	graphics mode
001	001	TV-graphics mode
111	000	user's mode

An instruction is another example of information stored in the packs and is stored in all of the channels of symbol 1. Corresponding mode, item, parity or additional information for the instruction is stored in all of the channels of symbols 2 to 7. Parity information for all of the data in the channels of symbols 0 to 19 is stored in all of the channels of the last 4 symbols (symbols 20 to 23) of each pack.

As discussed, the CD-G system is slow. The CD-G data is transmitted at a repeating frequency of 75 Hz and, therefore, a packet which contains 4 packs is transmitted at a rate of 300 packs per second (75 Hz x 4 packs). That is, with 1 character allocated to the range of 6 x 12 pixels, 300 characters can be transmitted in 1 second. However, a CD-G screen is defined as 288 horizontal picture elements x 192 CD-G vertical picture elements and requires more than twice the 300 characters transmitted in 1 second. The total transmission time for a 288 x 192 screen is 2.56 seconds as shown by the following equation:

$$(288/6) \times (192/12) \div 300 = 2.56 \text{ seconds}$$

With the CD-G system, searching for a specific event would be extremely time consuming because the time to regenerate each screen (2.56 seconds) by itself is extremely long, when it is considered that screens are usually refreshed in tenths of a second. This problem is compounded when hexadecimal codes are used for the characters because each hexadecimal expression requires 4 bits to represent 1 pixel. As a result, 4 times the data described above is transmitted, thereby increasing the transmission rate to 10.24 seconds (4 x 2.56 seconds). Since each screen requires a sluggish 10.24 seconds for transmission, a continual transmission of screens means that a lag time of 10.24 seconds is experienced when transmitting screens using the CD-G technique.

In one type of system (known as the CAPTAIN system), dot patterns, as well as character codes, represent the subtitles. This system, however, does not appear to be significantly better than the CD-G system and suffers from some of the same disadvantages. That is, both systems lack the capability to search for a specific event efficiently. In addition, these systems do not provide subtitles with sufficient resolution power in displaying the subtitles. The CD-G system designates only 1 bit for each pixel, and this binary pixal data creates undesired aliasing and flicker. The CAPTAIN system, for example, is developed for a 248 (horizontal picture elements) by 192 (vertical picture elements) display, i.e., a low resolution display, and not for high resolution video pictures of 720 x 480.

Illustrative embodiments of the invention seek to provide:

an encoding method and apparatus for encoding multiple types of data streams to be searched;
an encoding technique that enables a user to search for a particular subtitle;
a computer-readable memory for directing a computer to search the multiple types of data stream; and/or
a decoding method and apparatus for decoding multiple types of data streams to be searched.

Aspects of the invention are specified in the claims.

The present invention provides an encoding apparatus and method which encodes multiple types of data streams such that an event in the video broadcast represented by data in the data streams can be located by locating data streams of the same type as the data.

The present invention further provides a computer-readable memory for directing a computer to search for an event in the video broadcast represented by data in the data streams by searching data streams of the same type as the data to be searched.

The present invention further provides a decoding apparatus and method which decodes multiple types of data streams such that an event in the video broadcast represented by data in the data streams can be searched by searching data streams of the same type as the data to be searched.

A more complete appreciation of the present invention and many of its attendant advantages will be readily obtained by reference to the following illustrative description considered in connection with the accompanying drawings, in which:

Fig. 1 is a block diagram of an illustrative data decoding apparatus of the present invention;
Fig. 2 is a block diagram of the subtitle decoder depicted in Fig. 1;
Figs. 3a and 3B are tables of addresses according to an embodiment of the present invention;
Figs. 4A and 4B is a diagram depicting subtitle search operation in normal and trick playback modes;
Fig. 5 is a table of communications between the system controller of Fig. 1 and the controller of Fig. 2;
Fig. 6 is a table of parameters for the communications between components of Fig. 1 and Fig. 2;
Figs. 7A to 7C are signal diagrams demonstrating data encoding of an embodiment of the present invention;
Fig. 8 is a color look up table referred to when encoding subtitle data;
Figs. 9A and 9B constitute a block diagram of an illustrative encoding apparatus of the present invention;
Figs. 10A and 10B depict a block diagram for the wipe data sampler of Fig. 9A;
Fig. 11 is a color look up table referred to when conducting a color wipe operation;
Fig. 12 is a graph for the explanation of a code buffer operation;
Fig. 13 is a block diagram describing the internal operation of the code buffer in Fig. 2;
Figs. 14A to 14C depict a scheme for the colorwiping operation;
Fig. 15 is a block diagram depicting the colorwiping operation according to Figs 14A to 14C;
Figs. 16A to 16C depict a scheme for the dynamic positioning operation;
Fig. 17 is a block diagram depicting a circuit for the dynamic positioning operation according to Figs. 16A to 16C;
Figs. 18A to 18C depict the arrangement of data according to a CD-G format; and
Fig. 19 depicts a transmission format of the data in the CD-G format.

Decoding Apparatus

The illustrative data decoding apparatus shown in Fig. 1 in accordance with the present invention operates to decode a reproduced signal. A system controller 14 of the data decoding apparatus causes the reproduced signal to

be processed and sent to a subtitle decoder 7. The system controller communicates with a controller 35 (Fig. 2) of the subtitle decoder to decode the subtitles and superimpose them onto a decoded video image for display on a television screen.

A data decoder and demultiplexer 1 receives a digital signal reproduced from, for example, a VCR. The data decoder and demultiplexer error decodes the reproduced signal, preferably employing an Error Correcting Code (ECC) technique, and demultiplexes the error decoded reproduced signal into video, subtitle and audio data streams. A memory 2 may be used, for example, as a buffer memory as a work area for the purpose of error decoding and demultiplexing the reproduced signal.

A video decoder 3 decodes the demultiplexed video data from a video data stream. A memory 4 may be employed for the operation of decoding the video data similar to the operation of the memory 2 employed with data decoder and demultiplexer 1.

A letter box circuit 5 converts the decoded video data with a 4:3 aspect ratio to a 16:9 aspect ratio. The conversion is performed using a 4 to 3 decimation process, whereby every four horizontal lines are decimated to three horizontal lines, thus squeezing the video picture into a ¾ picture. According to the letter box format, a vertical resolution component is derived from the remaining ¼ of the video picture which is employed to enhance the vertical resolution of the decimated video picture. A timing adjustment memory 6 times the transmission of the video picture to ensure that the ¼ of the letter box picture is not transmitted. When the decoded video data generated by the video decoder 3 is already in a 16:9 letter box format, the letter box circuit bypasses the decimation operation and sends the decoded video data directly to subtitle decoder 7.

The decoded subtitle data demultiplexed by the data decoder and demultiplexer 1 is sent directly to subtitle decoder 7 which decodes the subtitle data according to instructions from system controller 14 and mixes the decoded subtitle data with the decoded video data.

A composite encoder 8 encodes the mixed subtitle data and video data into a suitable video picture format, such as NTSC, PAL or the like. A mode display 9 interfaces with a user and indicates, for example, the mode of a television monitor connected to the illustrated apparatus. A D/A converter 10 converts the encoded signal received from the composite encoder into an analog signal suitable for display in the indicated mode, such as NTSC or PAL.

The audio portion of the audio/video signal decoded by the data decoder and demultiplexer 1 is decoded by an audio decoder 11 which decodes the demultiplexed audio data using a memory 12, for example. The decoded audio data output from the audio decoder is converted into an analog audio signal appropriate for reproduction by a television monitor by a D/A converter 13.

Subtitle Decoder

Subtitle decoder 7, as will be discussed with reference to Fig. 2, decodes the encoded subtitle data and mixes the decoded subtitle data with the appropriate video data. Controller 35 (Fig. 2) controls the operations of the subtitle decoder and communicates with the system controller 14 of the decoder (Fig. 1) using the command signals shown in Fig. 2 (as listed in Fig. 5). Together, the controller 35 and system controller 14 time the decoding of the subtitle data so that the subtitle data is mixed with video image data at the proper position whereat the subtitles are to appear on the video image.

A word detector 20 of the subtitle decoder receives the subtitle data in groups of bit streams reproduced from a disk, the bit streams being stored on the disk in packets. Each group of bit streams makes up one frame (or page) of subtitles to be superimposed on a video image. Different groups of bit streams may represent subtitles displayed in different playback modes, such as normal playback, fast-reverse or fast-forward, alternatively referred to as trick modes. The system controller indicates to the word detector using a **stream_select** signal which playback mode is to be adopted for display and the word detector selects the appropriate bit stream of signals for the indicated playback mode. In the case where different video images are displayed on different channels, the system controller indicates the appropriate channel to the word detector correspondingly in a **ch_select** signal and the word detector changes channels to receive only those bit streams on the selected channel.

A group of bit streams making up one frame and received by the word detector includes header information (**s. header**) which describes the format of the group of bit streams. The header information is accompanied by header error information (**header error**) and data error information (**data error**). The system controller uses the header information to determine how to parse the group of bit streams and extract the relevant subtitle data therefrom. The system controller uses the header error information to correct anomalies in the header information and uses the data error information to correct anomalies in the subtitle data.

The word detector forwards the subtitle data (**Bitmap**) along with other decoded information (including a presentation time stamp **PTS**, position data **position_data** and color look up table data **CLUT_data**) to a code buffer 22. The **PTS** is a signal that indicates the precise time when the audio, video and subtitle data for a frame is transmitted so that the system controller knows when to demultiplex the data from the reproduced signal. The position data indicates

the horizontal and vertical position where the subtitles are to be superimposed on the video image. The **CLUT_data** indicates which colors are to be used for the pixels making up the subtitles. For example, the system controller 14 determines that a video image is being displayed and sends the subtitle data to subtitle decoder 7 at the time indicated by the time stamp (**PTS**) and causes the subtitle decoder to output the corresponding subtitle data (**Bitmap**) at a position in the video image represented by the horizontal and vertical position indicated by the **position_data** in the color indicated by the **CLUT_data**.

A scheduler 21 is provided to ensure that the data received by the code buffer 22 from the demultiplexer 1 (Fig. 1) does not overflow the code buffer. The scheduler controls read/write access to the code buffer by determining the bandwidth for an I/O port (not shown) which receives the bit streams selected by the word detector. The bandwidth refers to the read/write rate and is calculated by dividing the rate at which the demultiplexer demultiplexes data by the number of parallel bits written or read from the code buffer. For example, a data rate from the demultiplexer of 20 Mbps divided by 8 parallel bits results in a 2.5 Mbps rate of data read from the code buffer. Therefore, the scheduler will set the read/write rate of the I/O port in order to maintain a consistent flow rate of data into and out of the code buffer. The code buffer, thus, receives the subtitle data (**Bitmap**) and awaits a **decode start** signal from the system controller to read out the data.

Advantageously, the system controller executes reading in real time when it is determined from the horizontal and vertical sync signals that the television display is at a position corresponding to the position indicated by the **position_data**. For real time display, the reading rate should correspond to a picture element sampling rate, preferably 13.5 MHz. As discussed, the subtitle data preferably is written into the code buffer at a rate of 2.5 MHz or more. Thus, the 13.5 MHz sampling clock is divided into four clock cycles of 3.375 MHz each. One of these 3.375 MHz clock cycles is allocated to writing (because writing requires at least 2.5 MHz) and the remaining three clock cycles are allocated to reading data from the code buffer, thus satisfying the requirement for real time display.

The read/write operation described is not only advantageously performed in real time, but also provides high resolution. Eight bits of the subtitle data are read from the code buffer 22 for each of three read clock cycles, or twenty-four bits per sampling clock. When display of the picture is conducted by the television monitor every fourth clock cycle, one-fourth of the twenty-four bits, (24/4 =) 6 bits are displayed at every clock cycle. That is, each subtitle picture element may comprise six bits, which is more than sufficient to achieve a high quality of resolution for the subtitles.

The operation of the code buffer 22 and corresponding components of Fig. 2 is depicted in the block diagram in Fig. 13. The code buffer 22-1 accumulates bit streams of subtitle data until at least one page of subtitle data is accumulated in the code buffer. The subtitle data for one page is transferred from the code buffer 22-1 to the display memory 22-2 (which acts as a buffer for the subtitle decoder) when the subtitle portion of the display time stamp (PTS) is aligned with the synchronizing clock (SCR). The synchronizing clock advances a pointer in the display memory 22-2 during reading indicating which address of the stored subtitle data is being currently read. It will be noted that placing the code buffer and display memory in a single unit is preferred since the code buffer need only increment one pointer for pointing to the current address in the display memory 22-2 which stores the next set of subtitle data. With an internal memory, therefore, virtually no delay is attributed to a transfer operation, resulting in a high speed transfer of the subtitle data.

When the code buffer is read during a normal playback mode, the synchronizing clock advances the pointer of the display memory 22-2 at each clock pulse. However, during special (or trick) reproduction (such as fast-forward, fast-reverse playback modes), the pointer is advanced at a different rate. To this end, a **special** command is first sent to the controller 35 and the controller sends back an acknowledge signal **(special_ack),** acknowledging that special reproduction is to be initiated. To uniformly speed up (or slow down) the operations of the subtitle decoder according to the special reproduction rate, the system clock reference (SCR) can be altered by adding or subtracting clock pulses. Subtraction pulses are created at an n times rate corresponding to the rate of fast-feeding or fast-reverse feeding. For example, at the time when special reproduction is commenced, real time subtraction is performed on the bit stream of subtitle data read out from the code buffer at the n times rate and the pointer advances at the desired rate to effect the special playback mode. When the special reproduction operation corresponds to a pause operation, on the other hand, no subtraction pulses are created. Instead, an identical frame is continuously read from the code buffer repeatedly, thus providing the illusion sensation that the subtitles are paused.

The reading operation is ended when subtitle decoder 7 determines that an **end of page** (EOP) of the subtitle frame is reached. The system controller sends a **repeat time** signal to the controller 35 which indicates the length of a page. An inverse run-length circuit 24 includes a counter and sends a **display end** signal to the controller 35 when the count value of the counter reaches the value indicated by the **repeat time** signal. When controller 35 determines that the repeat time is reached, the reading operation of the code buffer is stopped. For purposes of this invention, the code buffer preferably stores at least two pages of subtitle data because one page will be read while another page is written into the code buffer.

Controller 35 issues a **buffer overflow** signal to system controller 14 when an overflow of code buffer 22 occurs. An overflow can be determined when the controller receives the **display end** signal from inverse run-length circuit 24 before word detector 20 receives an **end of page** (EOP) signal on the following page. At that time, the system controller

withholds transfer of subtitle data from data decoder and demultiplexer 1 (Fig. 1) to the word detector to prevent an overflow of the code buffer. When an overflow condition has passed, the next stream will be written into the code buffer and displayed at the correct display start position.

An underflow condition exists when code buffer 22 has completed reading the subtitle data for an entire page and no further data exists in the code buffer. The code buffer is depicted with a capacity of two pages by the "code buffer size" line in Fig. 12. Graphically, an underflow would appear in Fig. 12 as one of the vertical portions of line (C) which extends below the lower limit of the code buffer. By contrast, an overflow condition is graphically depicted in Fig. 12 when the subtitle data read into the code buffer is too large, i.e., the horizontal portion of line (C) extends beyond line (B).

Fig. 12 graphically demonstrates the data flow into and out of code buffer 22. The T-axis (abscissa) represents time, while the D-axis (ordinate) represents data size for each page of data. Thus, the gradient (rise/run) represents the data flow rate of the subtitles into the code buffer. Graph (C) represents the data flow of the subtitle data. The vertical portions of graph (C) indicate a transfer of subtitle data from the code buffer when the display time stamp (PTS) is aligned with the synchronizing clock (SCR) generated internally by subtitle decoder 7. The horizontal portions of the graph (C) indicate the transfer of subtitle data into the code buffer. For example, at a time that the display time stamp (PTS) for page (S0) is received by the code buffer, the previous page of subtitle data is transferred from the code buffer and page (S0) is written into the code buffer. When another display time stamp (PTS) is received by the code buffer, the subtitle data of page (S0) is transferred out of the code buffer and page (S1) is written in. Similarly, the remaining pages (S2), (S3) are written into and read out of the code buffer as indicated.

To precisely time the reading of the subtitle data from the code buffer with the display of the video image, delay compensation must be performed to allow for delays within the subtitle decoder. This is especially important where an external memory is employed as the display memory because an external memory increases the delay factor. Delay compensation is achieved by controlling the timing of the **decode start** command from system controller 14. The system controller delays the **decode start** command by a time equal to the processing of a letter box picture (approximately one field) and a delay caused by video decoding at the instant the synchronizing clock of the controller (SCR) is aligned with the display time stamp (PTS). Delay compensation is particularly useful, since the video, audio and subtitle data are multiplexed on the premise that the decode delay in each of the video, audio and subtitle data signals is zero in the data encoding apparatus.

When the subtitle data for one page is read out of the display memory 22-2 (Fig. 13), the headers of the bit streams are separated therefrom by a parser 22-3 and the remaining data is forwarded to the inverse variable-length coder or run-length decoder 23, 24 during a vertical blanking period (V). Inverse VLC (Variable Length Coding) circuit 23 (Fig. 2) subjects the subtitle data to variable length decoding. The variable length decoded subtitle data is composed of level data ("1" or "0") and run data as paired data. In the case where variable length decoding is not employed, the inverse VLC circuit may be bypassed and the subtitle data read from the code buffer will be directly output to inverse run-length circuit 24. Inverse run-length circuit 24 conducts run-length decoding by generating the level of data from the number of run data elements. Thus, VLC circuit 23 and run-length circuit 24 decompress the subtitle data which had been stored as compressed data in code buffer 22.

The decompressed subtitle data is then sent to a 3:4 filter 25. The 3:4 filter receives an xsqueeze signal from the system controller 14 indicating the aspect ratio of the corresponding television monitor. Where the signal indicates that the monitor has a 4:3 aspect ratio, the 3:4 filter applies 3:4 filtration processes to the subtitle data to match the size of the subtitles to the size of the video picture. In the preferred embodiment, the controller 35 reads 90 pixels worth of subtitle data from the code buffer 22 before the H sync pulse is generated. In the case where the television monitor already has a 16:9 aspect ratio, or the decompressed subtitle data represents fonts, the 3:4 filter is bypassed.

A color look-up table 26 (CLUT) receives the subtitle data from the 3:4 filter 25 and the **CLUT_data** from the code buffer 22. The color look up table generates a suitable color from the **CLUT_data** for the subtitle data. The color look up table selects an address corresponding to the subtitle data for each pixel and forwards a mixing ratio K and color components Y (luminance), C_R (color difference signal R-Y) and C_B (color difference signal B-Y) to the mixer 34. The color components Y, C_R and C_B, when mixed by mixer 34, at the mixing ratio K create a pixel with the color indicated by the color look up table.

Background video data is incorporated in the arrangement of the color look-up table. For example, address 0 of the look-up table includes key data K having the value of 00 h; which means that the subtitle data will not be seen and the background video data will manifest, as shown by regions T1 and T5 in Fig. 7c. Addresses 1h to 6h of the look-up table include values of the key data K which increase linearly (20, 40 ... C0 hexadecimal); which means that the subtitle pixels according to these addresses are mixed with the background data as shown by the regions T2 and T4 in Fig. 7c. Finally, addresses 8h to Fh of the look-up table include values of key data K of E0h; which means that the components Y, Cr and Cb are mixed without any background video data as shown by region T3 in Fig. 7c. The color look-up table data is generated from the system controller and is previously downloaded to the CLUT circuit before decoding. With the color look-up table, the filtered subtitle data is transformed into the appropriate color pixel for display on the television monitor.

Fig. 8 shows one example of a color look-up table where the components Y, Cr, Cb and K are arranged according to the addresses 0...F (hexadecimal). As will be explained, color wiping is performed by changing the **CLUT_data**, thereby replacing part of the color look up table by the color wiping color look up table shown in Fig. 11. Normally, a particular subtitle frame is refreshed several times because frames are refreshed in a television signal several times a second. When the subtitles are refreshed, the same subtitle data will be employed. However, the color will be different due to the changed color look up table. Thus, the subtitles will appear to be color wiped as they are refreshed with each consecutive frame.

A mixer 34 (Fig. 2) mixes the pixels from the color look-up table 26 with video data from video decoder 3 (Fig. 1). The resulting mixed data represents a video picture with superimposed subtitles and is ready to be output to a television monitor. Mixer 34 is controlled to position the subtitles within the video picture by referencing a **u_position** signal generated by system controller 14 from commands of an operator via controller 35. The **u_position** value designates the vertical position for display on the screen and may be varied (either by a user, the transmitter, or otherwise) allowing a user to place the subtitles anywhere along a vertical axis.

The illustrative decoding apparatus of the present invention may be practiced with the parameters for the different signals shown in Fig. 6. However, the present invention is not limited to the parameters set forth in that figure and may be employed in different video systems.

With the illustrative apparatus of the present invention, a user has control over the display of the subtitle through a mode display device 9 (Fig. 1). System controller 14, upon command from the user, sends a **control** signal to mixer 34 (Fig. 2), turning the subtitles on or off. Since the decoding apparatus decodes subtitles in real time, the user does not experience any unpleasant delay when turning the subtitles on or off. In addition, the subtitles can be controlled, by the user or otherwise, to fade-in/fade-out at a variable rate. This is achieved by multiplying a fade coefficient to the pattern data representing the subtitles at a designated speed. This function also allows an editor of the subtitles to present viewers with different sensations according to the broadcast of the audio/video picture. For example, news information may be "flashed" rapidly to draw the attention of the viewer, whereas subtitles in a slow music video "softly" appear in order not to detract from the enjoyment of the music video.

Decoding Multiple Types of Subtitles

The data decoding apparatus of Fig. 1 is designed to search a specific scene in the video image or sound. To that end, the apparatus decodes subtitles which indicate to a viewer a current event being displayed or speech being uttered during the search. The subtitle to be decoded, and eventually displayed, will be different for the indicated scene or sound searched. For example, the subtitle decoded may be a caption of the current event or a verse of speech. The apparatus determines which subtitles are decoded by distinguishing between the types of subtitles and decoding those which relate to the type of thing to be searched. It will appreciated that the other subtitles which do not relate to the thing searched are not decoded and displayed and, indeed, the subtitles used for searching are not necessarily displayed during normal reproduction.

The apparatus distinguishes between different subtitles by categorizing the subtitles into a multiple of stream data types (such as video, audio and subtitle types). As described with reference to word detector 20 (Fig. 2), the video data is stored on a record medium, such as disk 91, and therefore reproduced therefrom, as streams of data. Conceptually, these multiple types of data streams of data are shown in Figs. 4A, B with the video streams indicated by V_I, V_P, V_B, the audio streams by A, and the subtitle streams by Sw, Sp. The multiple types of data streams may be scattered on the disk and are reassembled by the decoding apparatus (Fig. 1) into coherent streams of information for television display.

The subtitles, thus, are also reproduced as streams of data and, as shown in Figs. 4A, B are different depending upon the playback mode, as shown by the subtitles Sp and Sw where are normal and trick mode subtitles, respectively. As with the video and audio data streams, the subtitle data streams may be scattered on the disk and, as shown in Figs. 4A, B, the subtitles are divided into three streams (Sp, Sw). The three streams are reassembled into one subtitle (Sp, Sw) in a forward playback mode by causing the pickup to jump over intermediary audio and video streams according to the arrows pointing from the left to the right in the figures (the subtitles in reverse playback are reassembled by following the arrows from right to left).

The decoding apparatus decodes only the video streams V_I during trick mode reproduction because the video streams V_I comprise the video frames corresponding to the I-frame in a series of MPEG (Motion Pictures Experts Group) frames which are not compressed on the basis of adjacent frames. On the other hand, the V_P and V_B video streams correspond to predictively encoded frames which are compressed on the basis of adjacent frames and decompression would require a longer period of time than allowed during fast-forward/reverse of a trick mode.

It will be appreciated that some video images do not have corresponding subtitle data because these video images are, for example, redundant or have little action. To reproduce these video images during the trick mode would cause a viewer to watch uninteresting images. In the apparatus, only the V_I streams have adjacent subtitle stream data Sw

and are decoded/displayed so that a viewer does not have to wait through video frames which have little interest. That is, only those frames which have significant activity, as predetermined upon encoding, will be assigned a subtitle and decoded for display. For example, a board room meeting scene where little activity takes place will be bypassed until an event of interest occurs and a subtitle will appear to alert the viewer of the interesting event. The viewer, then, commands the system controller to leave trick mode and enter normal playback mode, whereupon the trick mode subtitles indicating the events no longer are displayed.

Any scene in the video image or audible speech can be searched. To that end, the present apparatus distinguishes between multiple types (such as video, audio and subtitle types) of data streams (such as video, audio or subtitle data). Thus, for example, when a scene is searched, the system controller causes the subtitle decoder to decode only the subtitle data streams describing the video data. Similarly, when a sound is to be located, the system controller causes the subtitle decoder to decode only the subtitle data streams describing the audio data. The system controller operates by determining where the streams of the multiple types of data streams are located on the disk and causing the respective data streams for the type searched to be reproduced and sent to the decoder.

In a first embodiment of the invention, the system controller determines where the streams are on the disk by extracting a table of contents (TOC) from the beginning of the disk which provides the addresses for the multiple types of data streams. Figs. 3A, B depict a table of contents according to the present invention. The table of contents shown in Fig. 3A identifies frames by frame number (subcode frame #) and pointers (POINT) pointing to a track on the disk at which the corresponding frame is located, and time codes (PMIN, PSEC, PFRAME) corresponding to that frame. The table of contents of Fig. 3B identifies stream data according to the multiple types of streams by indicating the frame, starting sector address, and ending sector address of recorded data streams arranged by type (i.e., video, audio or subtitle). From this table of contents, the system controller can determine all of the streams in a particular type of stream data and cause drive control 15 (Fig. 1) to jump the pickup to the sector indicated by the start_sector address in the table of contents.

In a second embodiment of the invention, the sector addresses for the multiple types of streams are collected in an area of the disk called a stream map. Unlike the table of contents, the stream map is not necessarily confined to the beginning of the disk, but may be located at any sector as a packet. The stream map is, thus, arranged as a packet with video, audio, subtitle, blanking, packet length of stream, identifier, and length of stream map information. The system controller references the stream map in a similar manner to the table of contents, thus causing the pickup to reproduce subtitle streams corresponding to the type of data stream to be searched, for example, sending these reproduced streams to the subtitle decoder.

In a third embodiment, the sector addresses of the previous and following streams of a currently reproduced subtitle stream are stored in each subtitle stream. Since the sector addresses are in the subtitle streams, the sector addresses are decoded and word detector 20 of the subtitle decoder (Fig. 2) detects the subtitle stream sector addresses (**subtitle stream sector address**) and forwards them to the system controller, via controller 35. As each subtitle is decoded in, for example, the forward playback mode, the system controller recalls the following sector address for the next subtitle and causes the pickup to jump to the sector indicated by that following address to reproduce the next subtitle. Similarly, in the reverse playback mode, the system controller recalls the previous sector address for the previous subtitle and causes that subtitle to be reproduced. Specifically, the word detector detects whether a stream includes sector addresses according to the following operation:

	No. of bits	Mnemonic
user_data_flag	1	uimsbf
if(user_data_flag = "1")[		
length_of_user_data	16	bslbf
next_subtitle_address_offset	32	bslbf
reserved	8	bslbf
previous_subtitle_address_offset	24	bslbf
reserved	8	bslbf
]		

System controller 14 (Fig. 1) acts as a computer and executes the above operation, causing the word detector to determine if the user_data_flag is set to "1" and, if so, treats the next 16 bits as length_of_user_data; the next 32 bits as next_subtitle_address_offset; the next 8 bits as reserved; the next 24 bits as previous_subtitle_address_offset; and the last 8 bits as reserved. The word detector forwards this information to the system controller, via the controller 35, and continues detecting subtitle streams. The system controller receives the subtitle stream sector addresses and controls the decoding apparatus as described.

Encoding Technique

An illustrative encoding technique employed in an embodiment of the present invention will be described in more particular detail with reference to Figs. 7A, 7B, 7C and Fig. 8. As an example, the technique for encoding the letter "A" of Fig. 7A will be explained. The letter "A" is scanned along successive horizontal lines and the fill data of Fig. 7B is generated for the letter "A" along each horizontal line. It will be noted that the level "E0" demarks the highest level for recreating a color pixel from the color look-up table shown in Fig. 6, whereas level "0" represents a lack of subtitle data.

The key data (K) (or mixing ratio) determines the degree to which the fill data is mixed with background video. Regions T1 and T5 of the key data correspond to areas in the video picture that are not superimposed with the fill data; therefore, these areas are designated as level 0 as indicated by address 0 in Fig. 8. Regions T2 and T4 are mixed areas where the subtitles are gradually mixed with the background video picture so that the subtitles blend into the background video picture and do not abruptly contrast therewith. Any of the fill data in this area is stored in addresses 1 through 6 of the color look-up table. The main portion of the letter "A" is displayed within the T3 region where the background information is muted. The subtitle information in region T3 is stored as addresses 7 to F (hexadecimal). The color look-up table of Fig. 8 is arranged in varying degrees of the luminance component Y. When a pixel in the region T3 is to be stored, for example, and the level of the luminance component Y for that particular pixel is 20 (hexadecimal), the color information for that pixel is obtained from address 9 (Fig. 8). In this manner, the remaining pixels for the subtitle characters are encoded.

Encoding Apparatus

The illustrative encoding apparatus of the present invention is depicted in Figs. 9A, B. Audio and video information is received by a microphone 53 and video camera 51, respectively, and forwarded to a multiplexer 58. The subtitle data are entered through either a character generator 55 or a flying spot scanner 56 and encoded by a subtitle encoding circuit 57. The encoded subtitle information is sent to multiplexer 58 and combined with the audio/video information for recording onto a record disc 91 or supplied to a channel for transmission, display, recording or the like.

Video camera 51 generates the video signal and supplies the same to a video encoding unit 52 which converts the video signal from analog to digital form. The digitized video signal is then compressed for video transmission and forwarded to a rate controller 52a, which controls the rate that the compressed video data is transferred to the multiplexer in synchronism with the rate that the subtitles are sent to the multiplexer. In this manner, the compressed video data is combined with the subtitle data at the correct time. Similarly, audio information is obtained by microphone 53 and encoded by an audio encoding unit 54 before being sent to the multiplexer. The audio encoding unit does not necessarily include a rate controller because the audio data may ultimately be recorded on a different track or transmitted over a different channel from the video data.

The subtitles are generated by either character generator 55 or flying spot scanner 56. The character generator includes a monitor and a keyboard which allows an operator to manually insert subtitles into a video picture. The operator edits the subtitles by typing the subtitles through the keyboard. Flying spot scanner 56, on the other hand, is used for the situation where subtitles are already provided in an external video picture or scanned in as text. The flying spot scanner scans the video picture and determines where the subtitles are positioned and generates corresponding subtitle data therefrom. The subtitles from the flying spot scanner are pre-processed by a processing circuit 63 to conform with subtitles generated by the character generator before further processing by the subtitle encoding circuit.

The subtitle data from either character generator 55 or flying spot scanner 56 are, then, selected for compression. The character generator outputs blanking data, subtitle data and key data. The subtitle data and key data are forwarded to a switch 61 which is switched according to a predetermined timing to select either the subtitle or key data. The selected data from switch 61 is filtered by a filter 72 and supplied to another switch 62. Switch 62 switches between blanking data, the filtered data from the character generator, and the processed data from the flying spot scanner. When it is determined that no subtitles are present, the blanking data is chosen by switch 62. Where subtitles are present, switch 62 chooses between the character generator data or the flying spot scanner data, depending upon which device is being used to generate the subtitle data.

The data selected by switch 62 is quantized by a quantization circuit 64, using a quantization level based on data fed back from a subtitle buffering verifier 68. The quantized data, which may be compressed, is supplied to a switch 69 and (during normal operation) forwarded to a differential pulse code modulation (DPCM) circuit 65 for pulse code modulation. The modulated data is run-length encoded by a run-length coding circuit 66, variable-length encoded by a variable-length encoding circuit 67 and forwarded to the subtitle buffering verifier 68 for final processing before being sent to multiplexer 58.

Subtitle buffering verifier 68 verifies that the buffer is sufficiently filled with data without overflowing. This is done by feeding a control signal (referred to in Fig. 9A as a filter signal) back to the quantization circuit. The control signal changes the quantization level of the quantization circuit, thereby changing the amount of data encoded for a particular

subtitle. By increasing the quantization level, the amount of data required for the subtitle data is reduced and the bit rate of data flowing to the subtitle buffering verifier is consequently reduced. When the subtitle buffering verifier determines that there is an underflow of data, the control signal decreases the quantization level and the amount of data output from the quantization circuit increases, thereby filling the subtitle buffering verifier.

The subtitle buffering verifier is also responsible for preparing the subtitle data for transmission (over television airwaves, for example). The subtitle buffering verifier, to this end, inserts information necessary to decode the encoded subtitle data. This information includes a **normal/special play** signal which indicates whether the subtitles are recorded in a normal or special (fast-forward/reverse) mode (referred to above as the trick mode). An **upper limit value** signal is inserted which indicates the upper limit for the memory size of the subtitle data for a frame. An **EOP** signal marks the end of page for the subtitle data frame and also is inserted. A **time code** signal is inserted which is used as the time stamp **PTS** in decoding. **Subtitle encoding information** is inserted and includes information used in encoding the subtitle data, such as the quantization factor. **Positional information** is inserted and is used as the **position_data** upon decoding. A **static/dynamic** signal is inserted which indicates whether the subtitle data is in static or dynamic mode. The subtitle buffering verifier also inserts the color look up table address for transmission to the decoder so that the colors of the display will match the colors employed in creating the subtitles.

The subtitle buffering verifier is preferably a code buffer similar to the code buffer of the decoder (Fig.2). To that end, it is useful to think of the operation of the subtitle buffering verifier to be in symmetry (i.e., performing the inverse functions of the code buffer) with the code buffer. For example, the color pixels of the subtitles are converted into digital representations; the resultant digital subtitles are encoded by the run length encoder and the variable length encoder; header information is added; and the resultant subtitle information is stored in a buffer and forwarded to multiplexer 58 for multiplexing with the audio and video data.

Multiplexer 58 preferably employs time-sliced multiplexing; and also provides error correction processing (e.g., error correction coding) and modulation processing (e.g., EFM, eight-to-fourteen modulation). The multiplexed data is then transmitted (via television broadcasting, recording, or other means of transference) to the decoding apparatus for decoding and display.

Encoding Multiple Types of Subtitles

The present apparatus permits a viewer to search for a specific scene or audible speech by distinguishing between multiple types of data streams and selecting those data streams for display which are of the type relating to the search. For example, a video image is located by displaying the data streams of the subtitles describing the video image. The manner in which the encoder, already described with reference to Figs. 9A and B, encodes the multiple types of data streams will now be discussed. After multiplexer 58 multiplexes the audio, video and subtitle data streams, the multiplexed data is sent to a sectorizing processor 100 which arranges the data streams into fixed length sectors of packets. At this point, the data streams are ready for airwave transmission. When the data streams are to be recorded on a disk, however, a table of contents (TOC) & stream map generator 101 determines the addresses of the data streams to be recorded on the disk.

According to the first embodiment, the TOC & stream map generator generates the table of contents shown in Figs. 3A, B from the sectors generated by the sectorizing processor and the video/audio search information generated, for example, by a viewer. In the second embodiment, the TOC & stream map generator generates the stream map from the sectors generated by the sectorizing processor. Unlike the previous embodiment, the TOC & stream map generator inserts the stream map as a packet onto the disk. In the first two embodiments, the system controller 14 of the data reproducer (or receiver) reads the table of contents or the stream map directly and causes the decoding apparatus (Fig. 1) to decode the streams which relate to the data type being searched. In the third embodiment, the TOC & stream map generator inserts the stream addresses for the previous and following addresses into each of the subtitle streams. Unlike the first two embodiments, the system controller must cause the subtitle decoder to decode the subtitle stream and extract therefrom the sector addresses. As described in regard to decoding multiple types of streams, the TOC & stream map generator encodes each stream with 1 bit of a user_data_flag that indicates whether stream addresses are forthcoming in the stream; the next 16 bits as length_of_user_data; the next 32 bits as next_subtitle_address_offset; the next 8 bits as reserved; the next 24 bits as previous_subtitle_address_offset; and the last 8 bits as reserved.

In illustrative embodiments of the present invention, the video image, the audio track, and the subtitles are arranged in units of frames on the disk and the system controller accesses information by recalling from the disk the streams making up these frames. With this scheme, the system controller can cause the decoder to decode only those types subtitles which relate to the particular scene or sound to be searched. Thus, a viewer can browse through the video picture by reading the subtitles and view only those portions which are of interest. For that matter, the order or speed at which the information is reproduced can be altered freely. For example, if the viewer has auditory problems or is learning the language spoken on the audio track, the system controller can, under operator control, slow the rate of

reproduction for the audio track to allow the viewer to understand the reproduced speech.

Another application of the present invention relates to color wiping, where the information of color wiping changes frequency between frames. A viewer desiring to sing a portion of a Karaoke piece, for example, may search for a particular point in the song by viewing the subtitles and halting the search at the instant when the portion of the subtitles to be sung is highlighted by the colorwiping.

Colorwiping Encoding

Colorwiping refers to a process by which an image, such as the subtitles, is gradually overlayed with another image. An exemplary application of colorwiping is highlighting, wherein a frame of subtitles is dynamically highlighted from left to right with the passage of time. An embodiment of the present invention performs colorwiping by changing the color look up table at different points of time during the subtitle display. For example, an initial subtitle frame is generated with the standard color look up table in Fig. 8. When colorwiping is performed, the color look up table is changed to the color wiping look up table of Fig. 11. With the passage of each frame, the gradual change of the position at which the color look up table is changed from the colorwiping to the standard color look provides the sensation that the subtitles are changing color dynamically over time from left to right.

An encoding operation for color wiping will now be discussed with reference to Figs. 9A and 10. During the course of encoding subtitles, an operator may desire to color wipe the previously encoded subtitles. To that end, the operator is provided with a wipe lever 81 to control the colorwiping and a monitor 84 to view the color wiping in real time. The wipe lever is connected to an adapter 82 to adapt the analog voltages of the wipe lever to digital signals suitable for digital manipulation. The digital output of the adapter is fed to both a switcher 83 and a wipe data sampler 70. The switcher switches the color look up table to values represented by the position of the wipe lever and generates color pixels of the subtitles for display on the monitor. Thus, the operator can visually inspect the colorwiping procedure while it occurs and adjust the speed or color of the wiping to satisfaction.

The wipe data sampler and position sampler 70 determines from the adapter signals where in the video picture the color look up table is to be changed and outputs this information to encoding circuits 65, 66 and 67 (via switch 69) for encoding and transmission to multiplexer 58. Figs. 10A and 10B depict a block diagram of the operation of the wipe data and position sampler. A comparator 301 compares a present pixel signal generated by the adapter with a previous pixel signal from the adapter. This is achieved by transmitting the present pixel value to input A of comparator 301 while supplying the previous pixel value latched in a register 300 to input B of comparator 301. The comparator outputs a boolean "true" value to a counter 302 (which is reset at every horizontal or vertical sync pulse) when the present and previous pixels have the same value and, in response thereto, the counter increments a count value. That is, the comparator registers a true condition when the pixels up until that point are generated from the same color look up table. At the point where the color look up table changes, therefore, the present and previous pixels become unequal (i.e., their color changes) and the comparator generates a "false" boolean condition. The count value, thus, is equal to the number of matches between the present and previous values, which is the same as the position at which the color look up table changes. The count value is latched by a register 303 upon the following vertical sync pulse and transferred to the encoding circuits (via switch 69) for transmission.

Colorwiping Decoding

Color wiping decoding will now be discussed with reference to Figs. 14A-C and 15. Fig. 14A shows the position here the color look up table is switched at point **A** from a color wiping look up table (Fig. 11) to the standard color look up table (Fig. 8). Fig. 14B depicts a pattern of subtitle and colorwipe data arranged in discrete blocks of presentation time stamps **(PTS(n)** ... **PTS(n+t))**. The first presentation time stamp **PTS(n)** corresponds to normal subtitle data and the remaining presentation time stamps **PTS(n+1** ... **n+t)** correspond to colorwiping data **(WPA ... WPZ)**. Fig. 14C shows successive frames (**n ... n+t**) which correspond to the presentation time stamps. To execute colorwiping, each successive colorwiping frame (WPA ... WPZ) sets the point where the color look up table is switched (point A) further along the displayed subtitle, thereby dynamically performing colorwiping as a function of time.

An operational block diagram of the colorwiping decoding is depicted in Fig. 15. The vertical sync pulse triggers a register 205 to latch the current subtitle frame from a display buffer (Fig. 15 shows a colorwiping frame WP being latched). The colorwiping data latched by the register indicates the position of the color look up table switching. A pixel counter 208 decrements the value indicated by the colorwiping data at each horizontal sync pulse and outputs a boolean "true" flag to color look up table 26. While the flag is "true" the color look up table employs the colorwiping table (Fig. 11) to decode the colors of the subtitle pixels. When the pixel counter reaches zero, the position of color look up table switching is reached and the pixel counter issues a boolean "false" flag to color look up table 26. At this time, the color look up table switches the colorwiping color look up table (Fig. 11) to the standard look up table (Fig. 8), and the remainder of the subtitle frame is displayed in standard color mode. Each successive colorwiping frame (WPA ... WPZ)

moves the position of switching; thus, each refreshed subtitle frame advances (or retreats) the colorwiping, thus performing dynamic colorwiping.

The colorwiping color look up table in Fig. 11 incorporates two sets of colors (one set for addresses 0h to 7h and a second set for addresses 8h to Fh). Thus, the colorwiping color can be changed to a secondary color simply by changing the most significant bit (MSB) of the color look up table address. For example, the first set of colorwiping colors has a MSB of "0", while the second set has a MSB of "1". Changing the MSB of address 7h to a "1" transforms the address to Fh and the colorwiping color changes. This may be done, for example, by setting the MSB equal to the flag of pixel counter 208.

Employing the MSB to change between color sets has the advantage of reducing the number of bits required to be encoded. Since the MSB is known, only the three lower order bits need to be encoded where 4 bits are employed for every pixel. Where two bits are employed for every pixel, the subtitle data is coded only for the least significant bit. In a 4 bits per 2 pixel format, only the MSB is employed for color control and the remaining three bits can be reserved for pixel information. Thus, by using the MSB the number of bits encoded can be decreased and the overall processing time for encoding and decoding is optimized.

Dynamic Subtitle Positioning

The subtitles are repositioned dynamically, i.e., as a function of time, by employing a similar technique as described above with reference to colorwiping. As shown in Figs. 16A-C and 17 the position data is measured along the horizontal axis (Fig. 16A) and is transferred to the subtitle decoder with the subtitle data during the appropriate frame (Fig. 16C) corresponding to a presentation time stamp **(PTS(n),** for example; Fig. 16B).

The positioning operation will now be explained with reference to Fig. 17. The position data is a value representing the position of the subtitle frame along the horizontal axis and is read out from the display buffer and latched by register 205 on each vertical sync pulse. Pixel counter 208 decrements the position data on each horizontal sync pulse and send a boolean flag to controller 35 (Fig. 2) to indicate that the position of the subtitle frame has not been reached. When the pixel counter reaches zero, the position of the subtitle frame has been reached and the boolean flag is toggled to indicate this to the controller. The controller which has been delaying the reading operation of code buffer 22 (Fig. 2), then causes the code buffer to read out the subtitle data to run length decoder 24 (Fig. 2). The subtitle data is then decoded as described above and displayed with the corresponding video image. In this manner, the position of the subtitle frame is changed with each frame; thus providing dynamic movement of the subtitle frame.

It will be appreciated that the present invention is applicable to other applications, such as television or video graphics. It is, therefore, to be understood that, within the scope of the appended claims, the invention may be practiced otherwise than as specifically described herein.

Claims

1. An encoding apparatus for encoding data streams of multiple types including video, audio and subtitle types of data streams, onto a record medium, a sequence of the data streams for each type constituting a respective video, audio and subtitle signal which combine into a video broadcast, said encoding apparatus comprising:

   encoding means for encoding the video broadcast into data streams of multiple types to be recorded on the record medium;
   address generating means for generating addresses for each of the data streams pointing to a position where the respective data stream is recorded on the record medium; and
   address storing means for storing said addresses generated by said address generating means, according to the type of data stream to which they point, in areas of the record medium reserved by type of data stream,
   wherein, an event represented by data in the data streams can be located by accessing the data streams of the same type as the data to be located.

2. The encoding apparatus of claim 1, wherein subtitles are generated from the recorded data streams to be displayed exclusively during a trick data playback mode,

   wherein said encoding means encodes said subtitles exclusively generated for trick mode as data streams on said record medium,
   wherein said address generating means generates addresses for said subtitles exclusively generated for trick mode, and
   wherein said address storing means stores the addresses of the subtitles exclusively generated for trick mode

at a position on the record medium indicating that the addresses are of the subtitle type exclusively generated for trick mode such that said event represented by data can be located by accessing said data streams of said subtitles exclusively generated for trick mode.

3. The encoding apparatus of claim 2, wherein the address storing means further comprise table of contents means for storing the addresses in a table according to the type of recorded data stream in an area of the record medium preceding an area where the data streams are stored.

4. The encoding apparatus of claim 2, wherein the address storing means further comprise stream map means for storing the addresses according to the type of recorded data stream in a packet of data streams recorded on the recording medium.

5. The encoding apparatus of claim 2, wherein the address storing means further comprise means for storing, in each of the data streams of the subtitles exclusively generated for trick mode, a next address of the following data stream and a previous address of the preceding data stream in the sequence for the data stream of subtitles.

6. An encoding method for encoding data streams of multiple types, including video, audio and subtitle types of data streams, onto a record medium, a sequence of the data streams for each type constituting a respective video, audio and subtitle signal which combine into a video broadcast, said encoding apparatus comprising:

encoding the video broadcast into data streams of multiple types to be recorded on the record medium;
generating addresses for each of the data streams pointing to a position where the respective data stream is recorded on the record medium; and
storing said addresses generated by said address generating means, according to the type of data stream to which they point, in areas of the record medium reserved by type of data stream,
wherein, an event represented by data in the data streams can be located by accessing the data streams of the same type as the data to be located.

7. The encoding method of claim 6, wherein subtitles are generated from the from recorded data streams to be displayed exclusively during a trick mode; and further comprising the steps of:

encoding said subtitles exclusively generated for trick mode as data streams on said record medium,
generating addresses for said subtitles exclusively generated for trick mode, and
storing the addresses of the subtitles exclusively generated for trick mode at a position on the record medium indicating that the addresses are of the subtitle type exclusively generated for trick mode such that said event represented by data can be located by accessing said data streams of said subtitles exclusively generated for trick mode.

8. The encoding method of claim 7, wherein the step of storing the addresses of the subtitles exclusively generated for trick mode comprises storing the addresses in a table of contents according to the type of recorded data stream into an area of the record medium preceding an area where the data streams are stored.

9. The encoding method of claim 7, wherein the step of storing the addresses of the subtitles exclusively generated for trick mode further comprises the step of grouping the addresses according to the type of data stream in a packet of data streams on the recording medium.

10. The encoding method of claim 7, wherein the step of storing the addresses of the subtitles exclusively generated for trick mode further comprises the step of storing, in each of the data streams of the subtitles exclusively generated for trick mode, a next address of the following data stream and a previous address of the preceding data stream in the sequence for the data stream of subtitles.

11. A computer-readable memory for directing a computer to search for data representing an event in a video broadcast stored in a data stream among data streams of multiple types including video, audio and subtitle types of data streams, on a record medium, a sequence of the data streams for each type constituting a respective video, audio and subtitle signal which combine into a video broadcast, comprising:

storing means for storing addresses pointing to said data streams of multiple types, wherein said storing means stores said addresses in groups indicative of the type of stored data stream; and

subtitle data pointing means for directing said computer during said search to sequentially decode said data streams positioned at addresses pointing to a subtitle type of data stream exclusively displayed during said search.

12. The computer-readable memory according to claim 11, wherein said storing means further comprise table of contents means for storing said addresses in a table arranged by the type of data.

13. The computer-readable memory according to claim 11, wherein said storing means further comprise stream map means for storing the addresses by the type of data stream in a packet on the recording medium.

14. The computer-readable memory according to claim 11, wherein the storing means further comprise means for storing, in each of the data streams of the subtitles exclusively displayed during said search, a next address of the following data stream and a previous address of the preceding data stream in the sequence for the data stream of subtitles.

15. A decoding apparatus for decoding data streams of multiple types, including video, audio and subtitle types of data streams reproduced from a record medium, a sequence of the data streams, for each type constituting a respective video, audio and subtitle signal which are combined into a video broadcast, said decoding apparatus comprising:

decoding means for decoding the multiple types of data streams reproduced from the record medium; and
control means for controlling said decoding means to decode data streams of the subtitle type which are to be displayed exclusively during a search, such that any event represented by data in the data streams is locatable by accessing the data streams of the same type as the data to be searched.

16. The decoding apparatus of claim 15 wherein the control means comprises means for reproducing a table of contents recorded on the record medium at a position preceding the data streams recorded thereon, said table of contents including addresses of the data streams of the subtitle data type which are to be displayed exclusively during said search.

17. The decoding apparatus of claim 15 wherein the control means comprises means for reproducing a stream map from a packet of the data streams recorded on the record medium, said stream map including addresses of the data streams of the subtitle data type which are to be displayed exclusively during said search.

18. The decoding apparatus of claim 15 wherein the decoding means decodes data streams reproduced from addresses of the record medium and comprises means for reproducing from the record medium a next address of the following data stream and a preceding address of the previous data stream of the sequence in each data stream is to be displayed exclusively during said search.

19. A decoding method for decoding data streams of multiple types, including video, audio and subtitle types of data streams, reproduced from a record medium, a sequence of the data streams for each type constituting a respective video, audio and subtitle signal which are combined into a video broadcast, said decoding method comprising the steps of:

decoding the multiple types of data streams reproduced from the record medium; and
controlling said decoding to decode data streams of the subtitle type which are to be displayed exclusively during a search, such that any event represented by data in the data streams is locatable by accessing the data streams of the same type as the data to be searched.

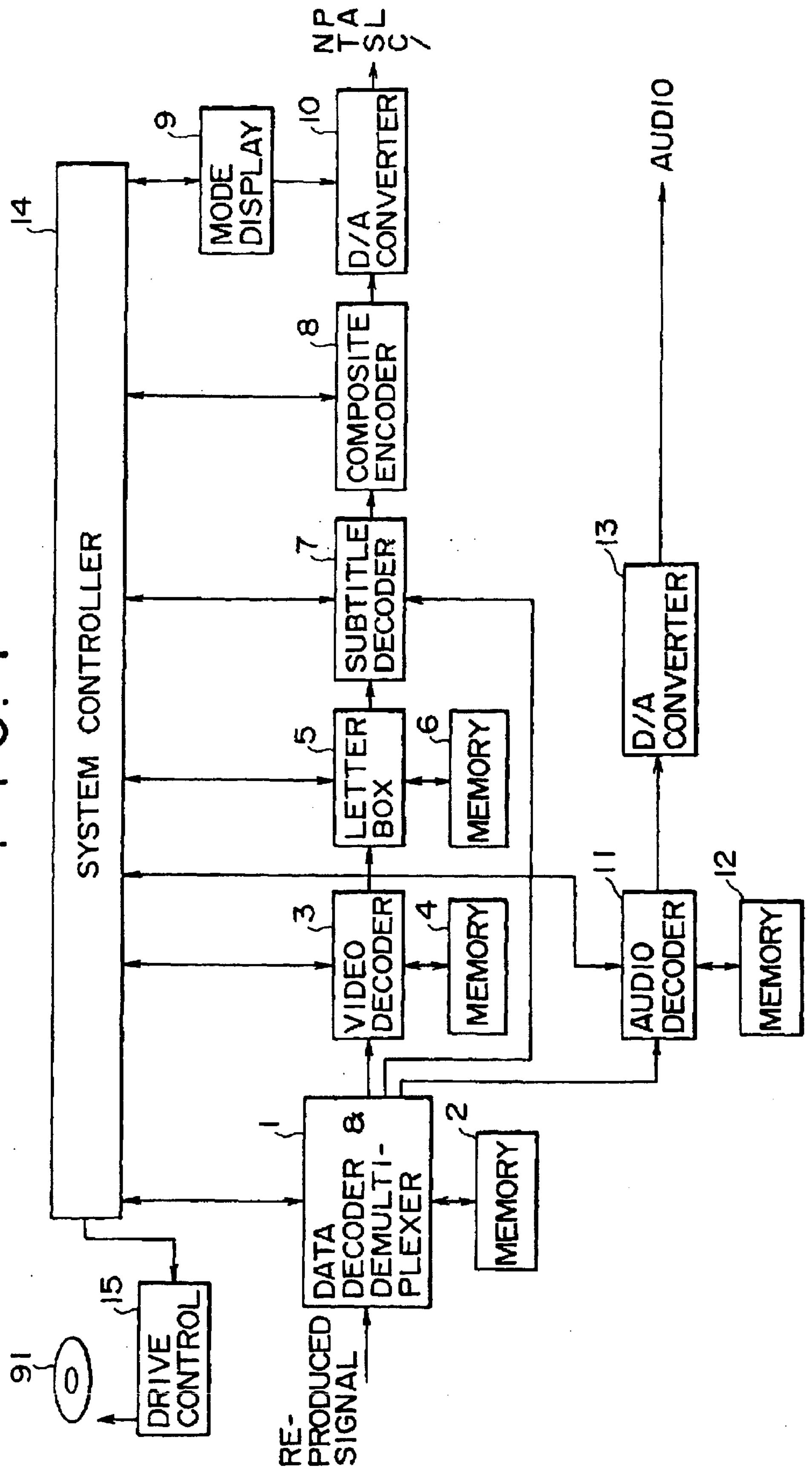
20. The method of claim 19 wherein the controlling step comprises the step of reproducing a table of contents recorded on the record medium at a position preceding the data streams recorded thereon, said table of contents including addresses of the data streams of the subtitle data type which are to be displayed exclusively during said search.

21. The method of claim 19 wherein the controlling step comprises the step of reproducing a stream map from a packet of the data streams stored on the record medium, said stream map including addresses of the data streams of the subtitle data type which are to be displayed exclusively during said search.

22. The method of claim 19 wherein the decoding step decodes data streams reproduced from addresses of the record

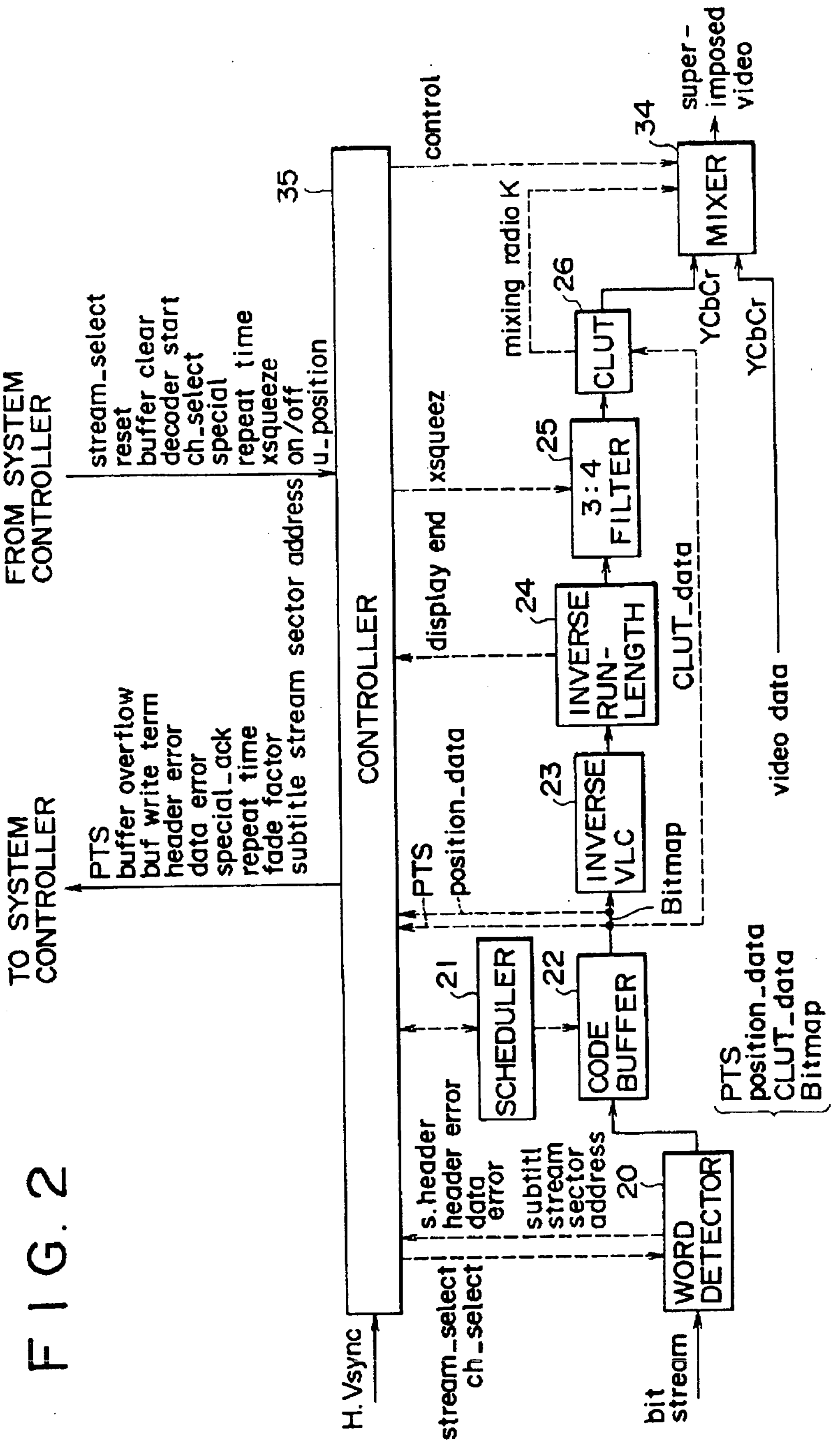
medium and comprises the step of reproducing from the record medium a next address of the following data stream in the sequence and a preceding address of the previous data stream in the sequence in each data stream to be displayed exclusively during said search.

FIG. 1



BLOCK DIAGRAM OF INTRA-PLAYER CIRCUITS

FIG. 2



SUBTITLE DECODER 7

FIG. 3A

TABLE OF CONTENTS IN CD

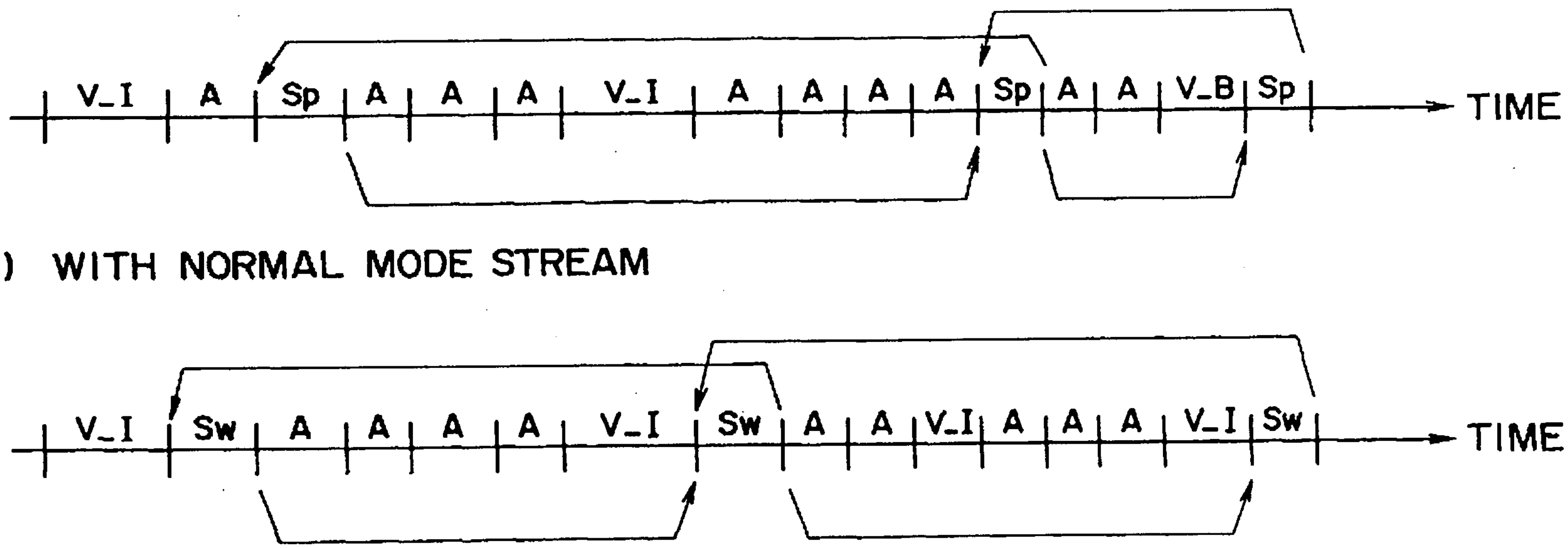
subcode frame #	POINT	PMIN, PSEC, PFRAME
N	xxxxxxxx	xx, yy, zz
N+1	xxxxxxxx	xx, yy, zz
N+2	xxxxxxxx	xx, yy, zz
N+3	xxxxxxxx	xx, yy, zz

FIG. 3B

TABLE OF CONTENTS

STREAM	FRAME	START_SECTOR_ADDRESS	END_SECTOR_ADDRESS
video	V	xxxxxxxx	xxxxxxxx
	V+1	xxxxxxxx	xxxxxxxx
	V+2	xxxxxxxx	xxxxxxxx
	V+3	xxxxxxxx	xxxxxxxx
audio	A	xxxxxxxx	xxxxxxxx
	A+1	xxxxxxxx	xxxxxxxx
	A+2	xxxxxxxx	xxxxxxxx
subtitle	S	xxxxxxxx	xxxxxxxx
	S+1	xxxxxxxx	xxxxxxxx
	S+2	xxxxxxxx	xxxxxxxx

FIG. 4



(A) WITH NORMAL MODE STREAM

(B) WITH TRICK MODE STREAM

V_I : INTRA VIDEO PICTURE
V_P : DIFFERENTIAL PICTURE
V_B : BIDIRECTIONAL DIFFERENTIAL PICTURE
A : AUDIO
Sw : SUBTITLE WHOLE PAGE
Sp : SUBTITLE DIVIDED PAGE

FIG. 5

	bits		
(1) FROM SYSTEM CONTROLLER 14			
reset	1	SYSTEM RESET	
buffer clear	1	INSTRUCTION FOR DELETING THE DATA FROM SYSTEM CONTROLLER BECAUSE THE CODE DATA HAS AN ERROR	max30Hz
decode start	1	START OF DECODING (START OF CODE BUFFER READ OPERATION)	max30Hz
stream_select	5	SPECIFYING THE STREAM INCLUDING IDENTIFICATION OF NORMAL/ SPECIAL PLAYBACK	static
ch_select	5	SPECIFYING A DECODING CHANNEL	static
special	1	SPECIAL PLAYBACK	as it happens
repeat time	8	DISPLAYING SPECIAL PLAYBACK TIME	as it happens
xsqueeze	1	USING 16:9 MONITOR	static
on/off	1	SUBTITLE SUPERIMPOSITION ON/OFF	static
u_position	8	DISPLAYING THE USER-SPECIFIED POSITION (VERTICAL DIRECTION ON THE SCREEN)	static
(2) TO SYSTEM CONTROLLER 14			
PTSS	33	TIME STAMP OF SUBTITLE DISPLAY TIME	max30Hz
buffer overflow	1	THERE ARE DATA OF TWO BANKS IN THE BUFFER	max30Hz
buf write term	1	COMPLETING WRITING OPERATION OF DATA OF ONE BANK	max30Hz
header error	1	THERE IS AN ERROR IN THE HEADER	max30Hz
data error	1	THERE IS AN ERROR IN THE DATA	max30Hz
special_ack	1	ACK FOR SPECIAL PLAYBACK	as it happens
repeat	8	DISPLAY TIME (FOR BOTH NORMAL AND SPECIAL PLAYBACKS)	max30Hz
v. position	8	DISPLAY POSITION DURING ENCODING OPERATION	max30Hz
fade factor	4	FADE IN/OUT TIME	max30Hz

FIG. 6

(1)(2):8bit bus+4bit select+1bit I/O
others:real signal bits

(3) from generator		bits
	H sync	1
	V sync	1
	13.5Mbz clock	1
(4) from demux		
	data stream	8
	strobe	1
	error	1
(5) to code buffer		
	address	15
	data	8
	xce	1
	xwe	1
	xoe	1
(6) from video decoder		
	video data(4:2:2)	16
(7) to DAC		
	video data(4:2:2)	16

FIG. 7A

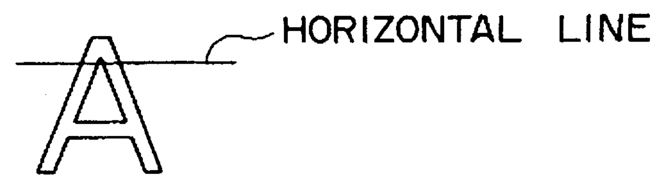


FIG. 7B

FIG. 7C

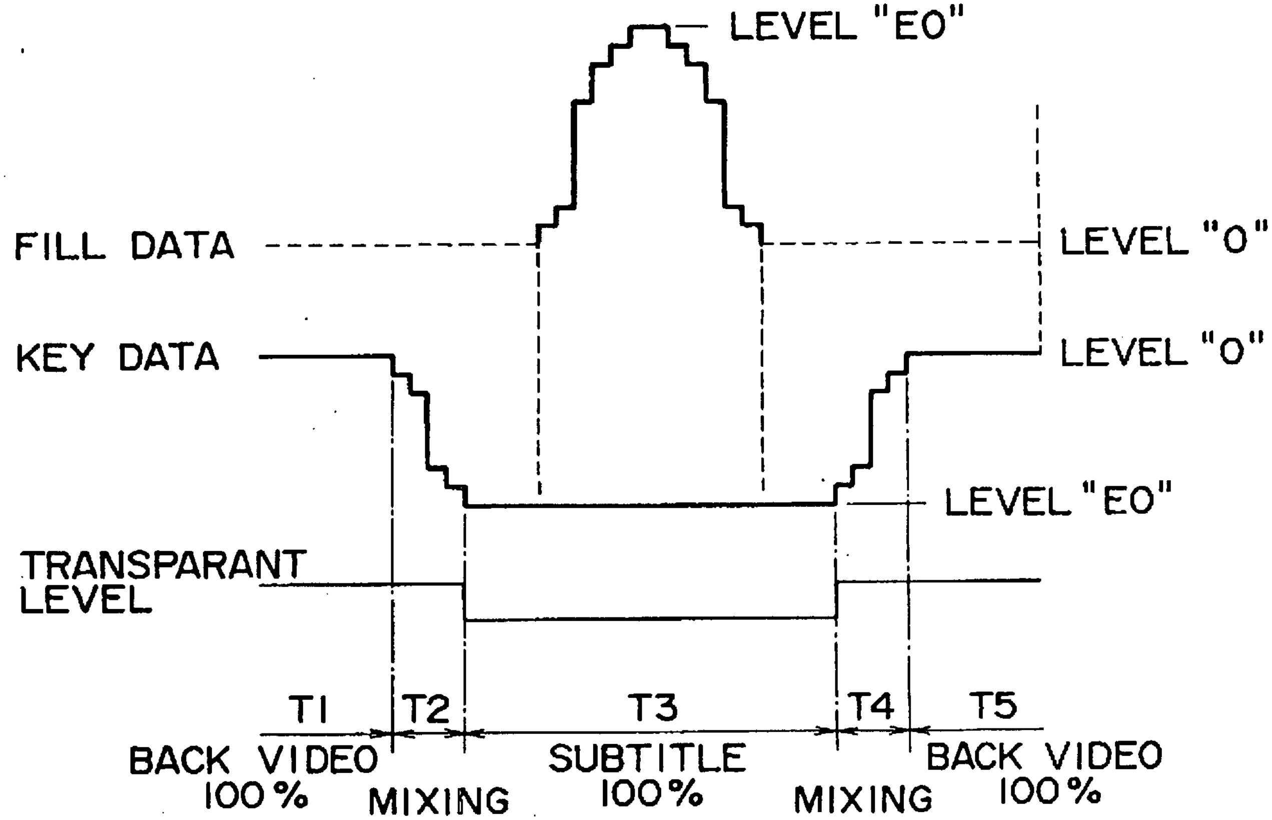


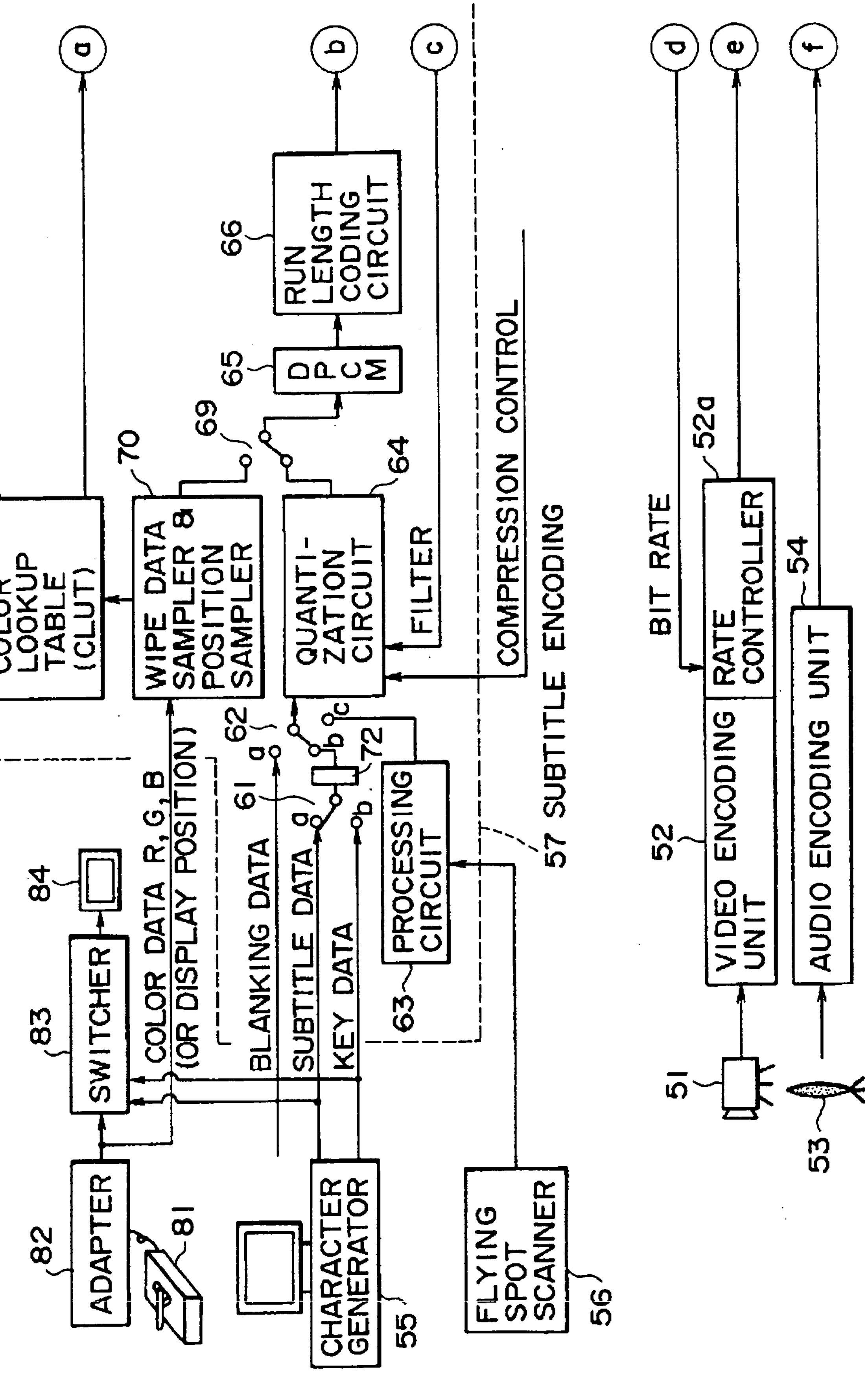
FIG. 8

Addr	Y	Cr	Cb	K
0	00	7F	7F	00
1	00	7F	7F	20
2	00	7F	7F	40
⋮				
6	00	7F	7F	C0
7	00	7F	7F	E0*
8	00	7F	7F	E0
9	20	7F	7F	E0
⋮				
E	C0	7F	7F	E0
F	E0	7F	7F	E0

* E0 : SUBTITLE DATA 100 %

: VIDEO DATA 0%

FIG. 9A



ADAPTER
82
81
SWITCHER
83
84
COLOR DATA R, G, B
(OR DISPLAY POSITION)
BLANKING DATA
SUBTITLE DATA
KEY DATA
CHARACTER GENERATOR
55
FLYING SPOT SCANNER
56
PROCESSING CIRCUIT
63
61
62
72
COLOR LOOKUP TABLE (CLUT)
71
WIPE DATA SAMPLER & POSITION SAMPLER
70
69
QUANTI-ZATION CIRCUIT
64
FILTER
COMPRESSION CONTROL
DPCM
65
RUN LENGTH CODING CIRCUIT
66
57 SUBTITLE ENCODING
BIT RATE
VIDEO ENCODING UNIT
52
RATE CONTROLLER
52a
AUDIO ENCODING UNIT
54
51
53
a
b
c
d
e
f

F I G. 9B

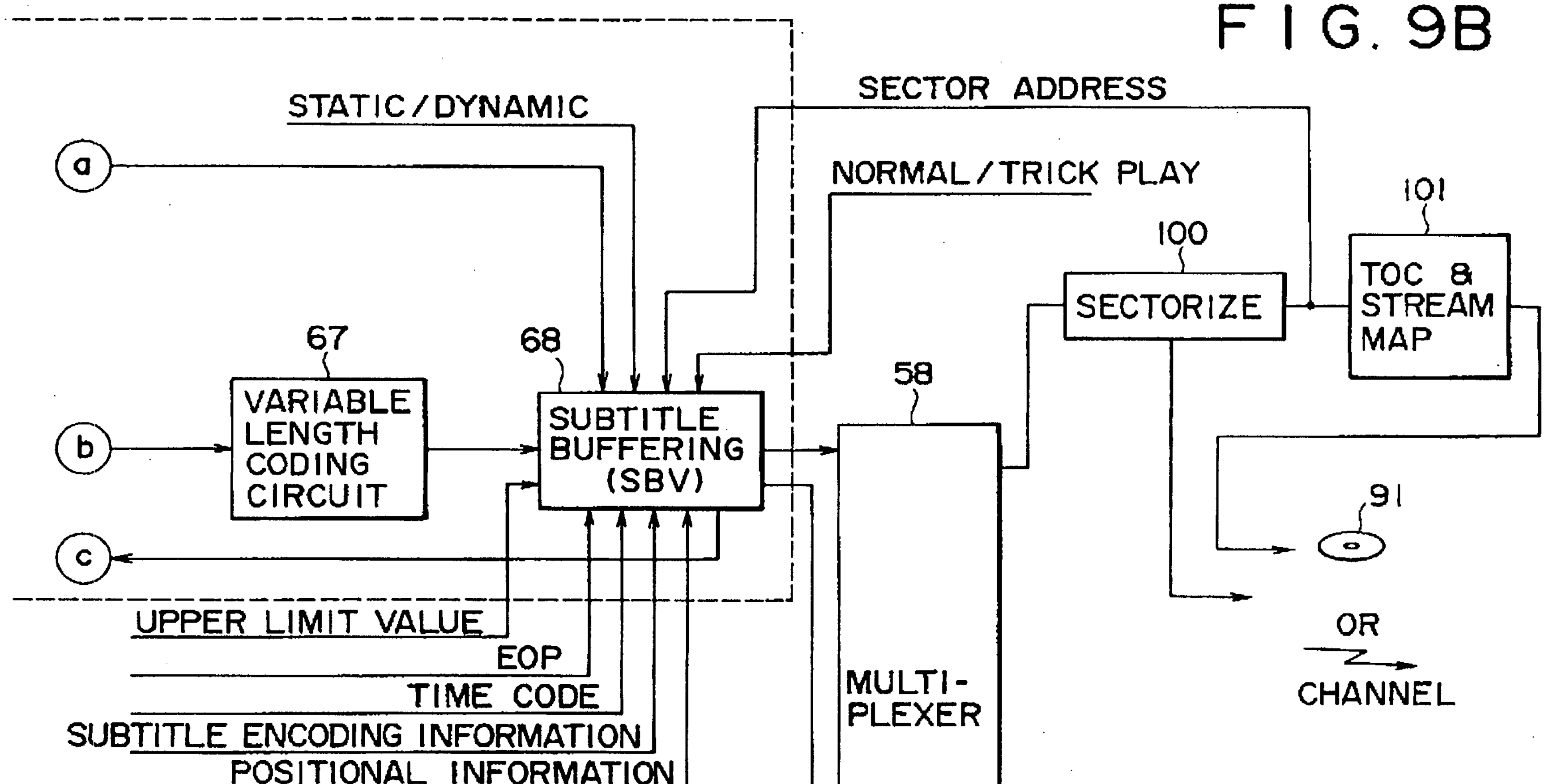

STATIC/DYNAMIC
SECTOR ADDRESS
NORMAL/TRICK PLAY
a
b
c
d
e
f
67
VARIABLE LENGTH CODING CIRCUIT
68
SUBTITLE BUFFERING (SBV)
58
MULTI-PLEXER
100
SECTORIZE
101
TOC & STREAM MAP
91
OR
CHANNEL
UPPER LIMIT VALUE
EOP
TIME CODE
SUBTITLE ENCODING INFORMATION
POSITIONAL INFORMATION

FIG. 9

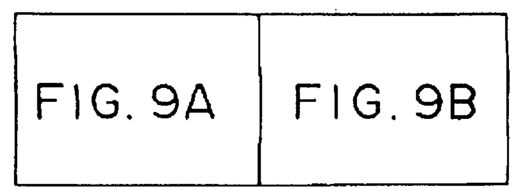


FIG. 9A
FIG. 9B

FIG. 10

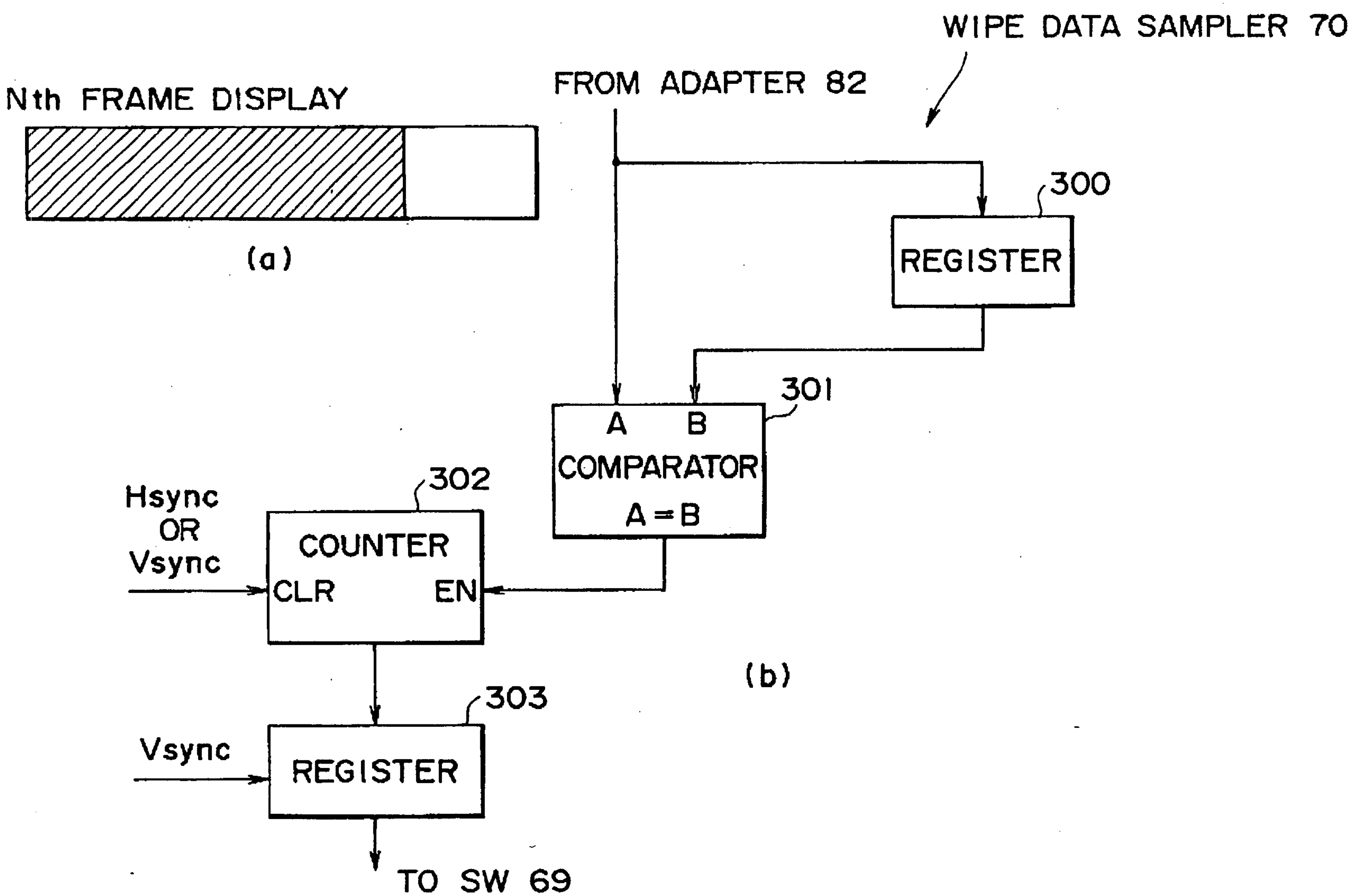

Nth FRAME DISPLAY
(a)
FROM ADAPTER 82
WIPE DATA SAMPLER 70
300
REGISTER
301
A
B
COMPARATOR
A = B
302
COUNTER
CLR
EN
Hsync
OR
Vsync
303
REGISTER
Vsync
TO SW 69
(b)

FIG. 11

COLOR LOOKUP TABLE

Addr	Y	Cr	Cb	K
0	00	7F	7F	00
1	20	7F	7F	40
2	40	7F	7F	80
3	60	7F	7F	C0
4	80	7F	7F	F0
5	A0	7F	7F	F0
6	C0	7F	7F	F0
7	E0	7F	7F	F0
8	00	FF	FF	00
9	20	FF	FF	40
A	40	FF	FF	80
B	60	FF	FF	C0
C	80	FF	FF	F0
D	A0	FF	FF	F0
E	C0	FF	FF	F0
F	E0	FF	FF	F0

FIG. 12

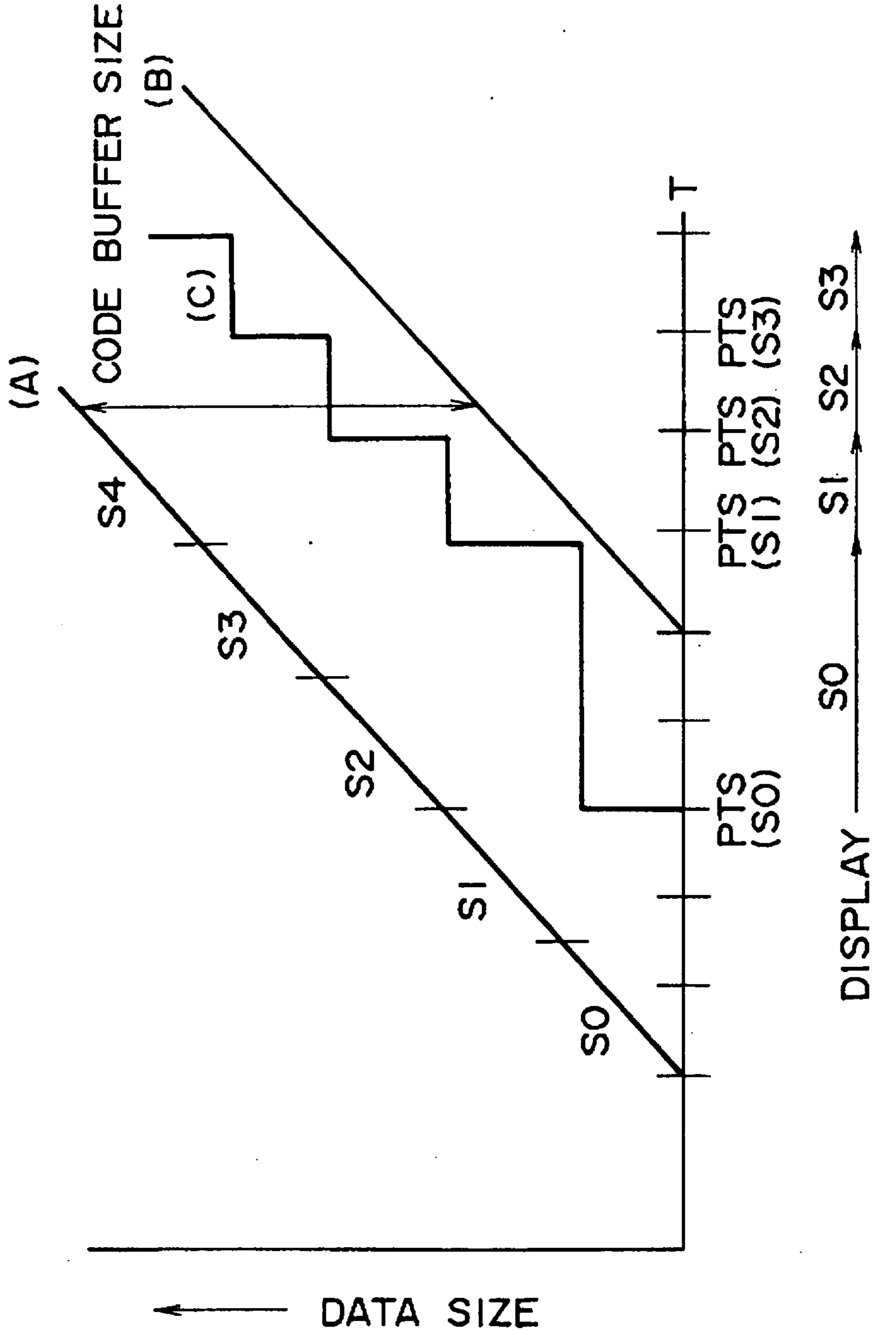

CODE BUFFER SIZE
(A)
(B)
(C)
S0
S1
S2
S3
S4
T
PTS (S0)
PTS (S1)
PTS (S2)
PTS (S3)
DISPLAY
S0
S1
S2
S3
DATA SIZE

FIG. 13

SUBTITLE DECODER BUFFER MODEL

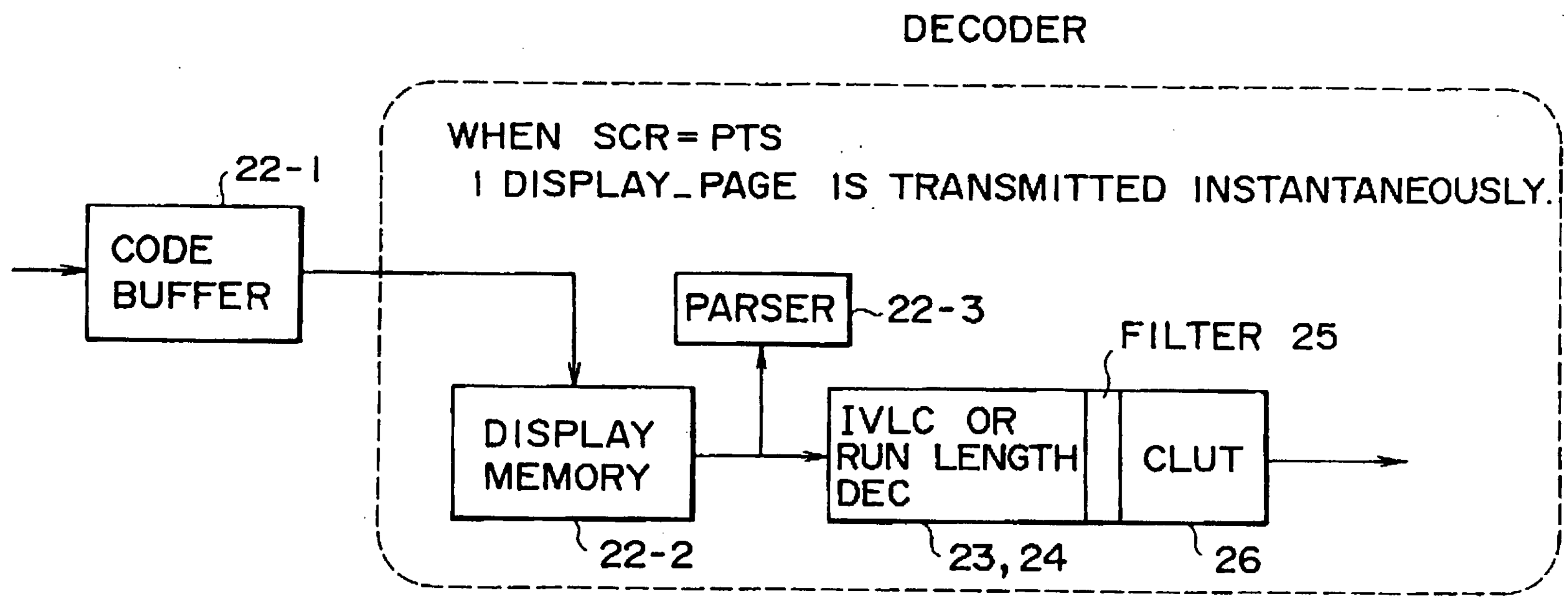
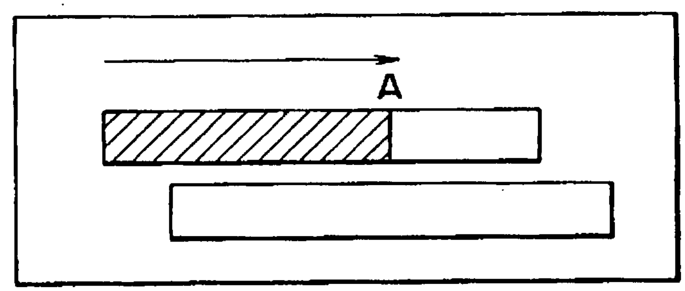


FIG. 14A



A: CLUT SWITCHING POINT

FIG. 14B

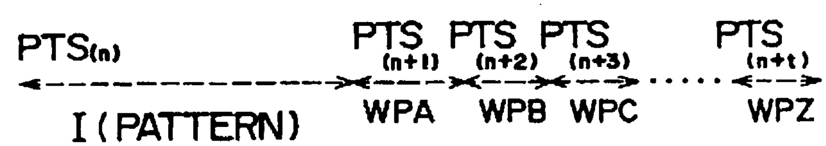
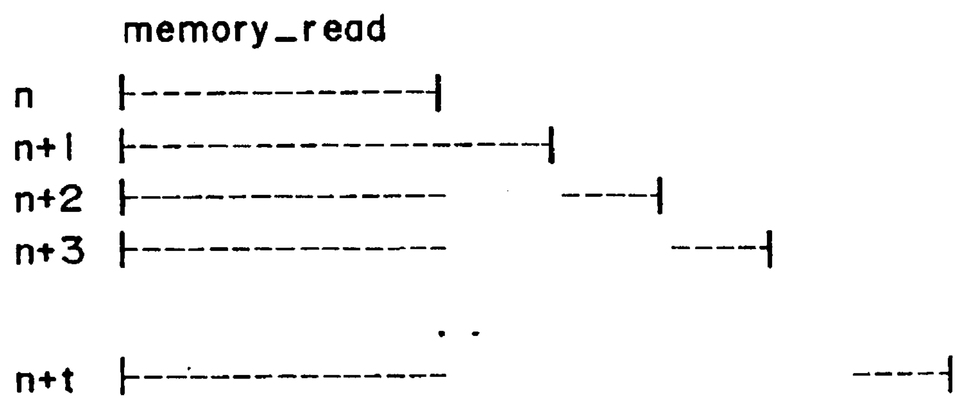


FIG. 14C



(t : FRAME FREQUENCY)

DISPLAYING PERIOD : (n+t)−n

FIG. 15

BLOCK DIAGRAM FOR CLUT SWITCHING

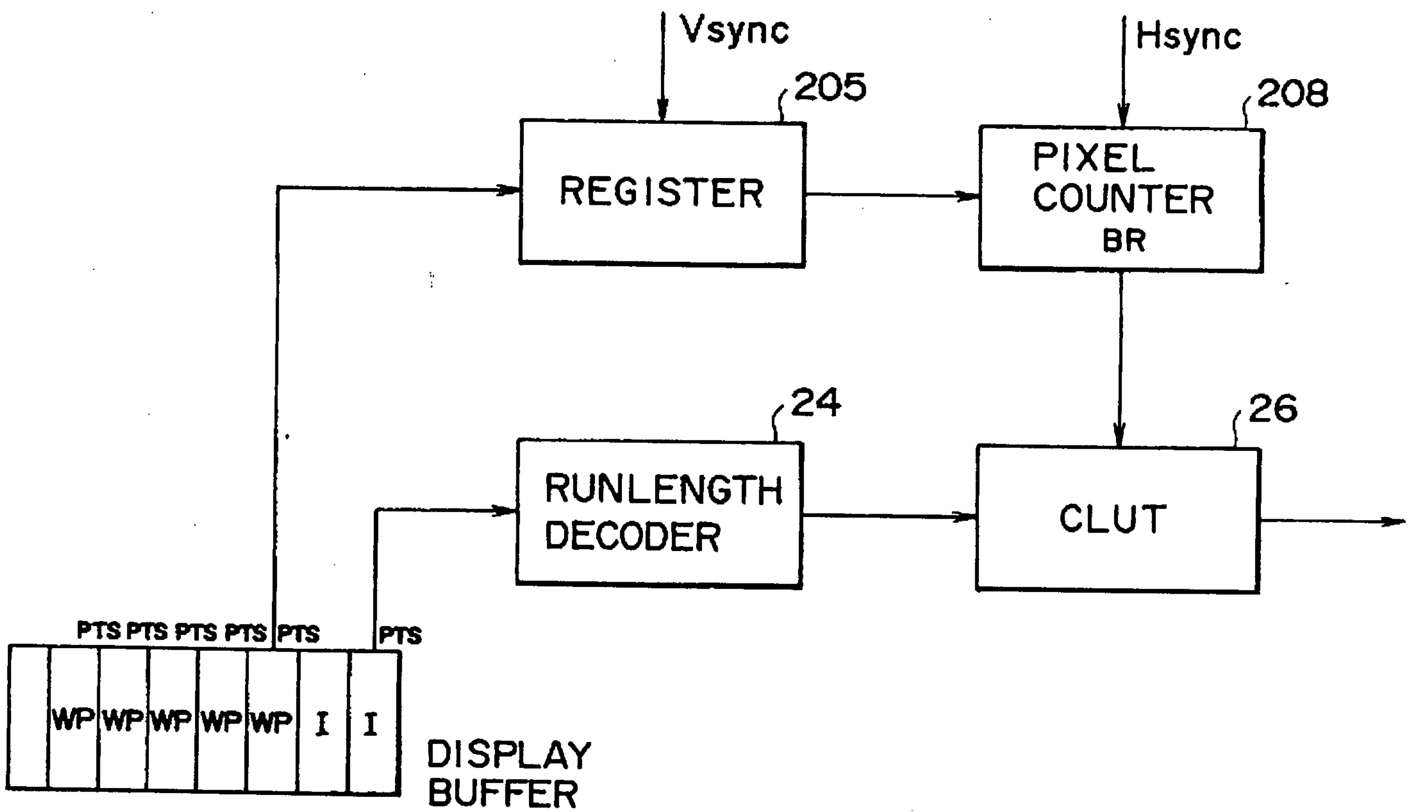


FIG. 16A

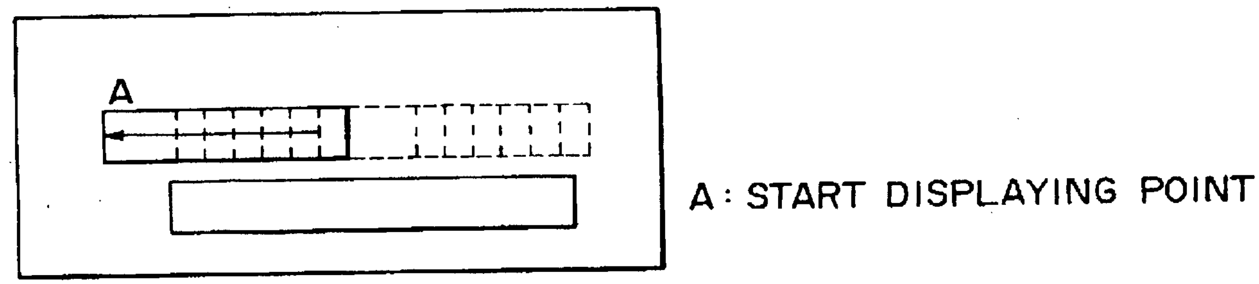


FIG. 16B

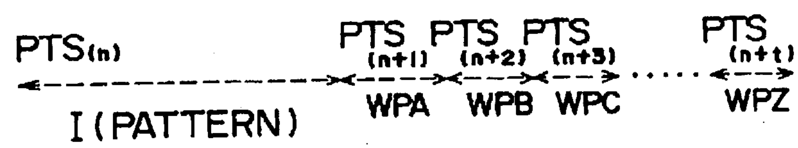


FIG. 16C

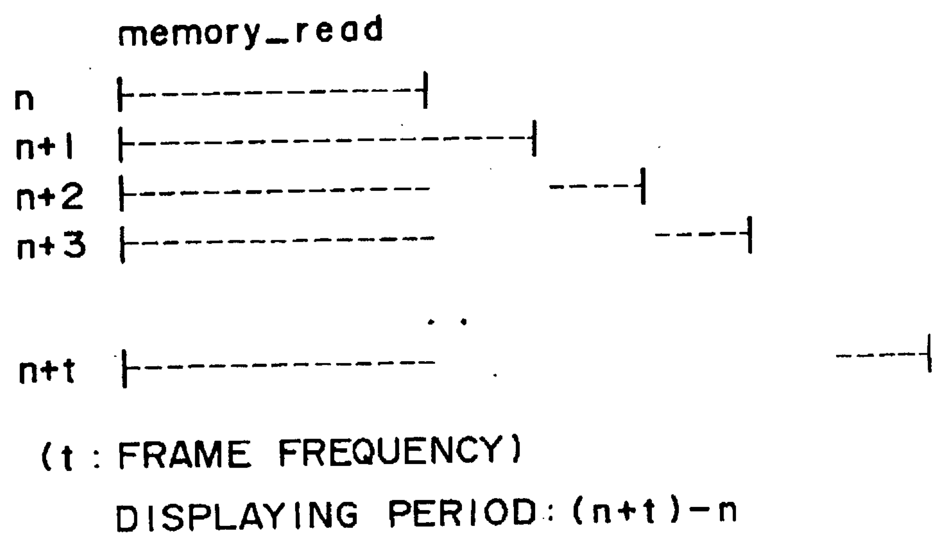


FIG. 17

BLOCK DIAGRAM FOR POSITION CONTROL

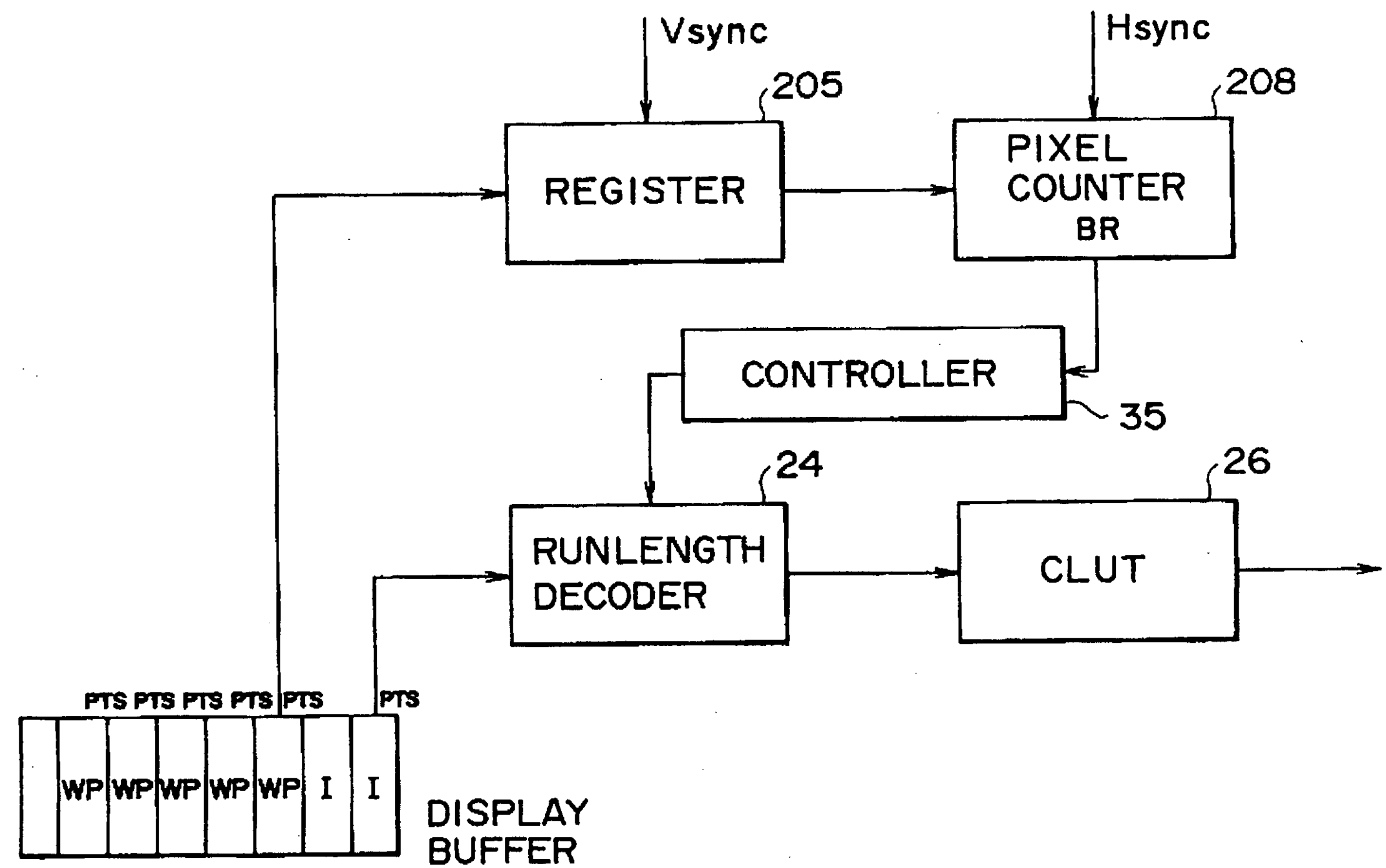


FIG. 18A

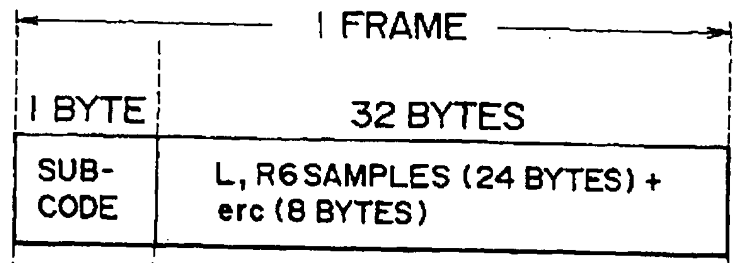


FIG. 18B

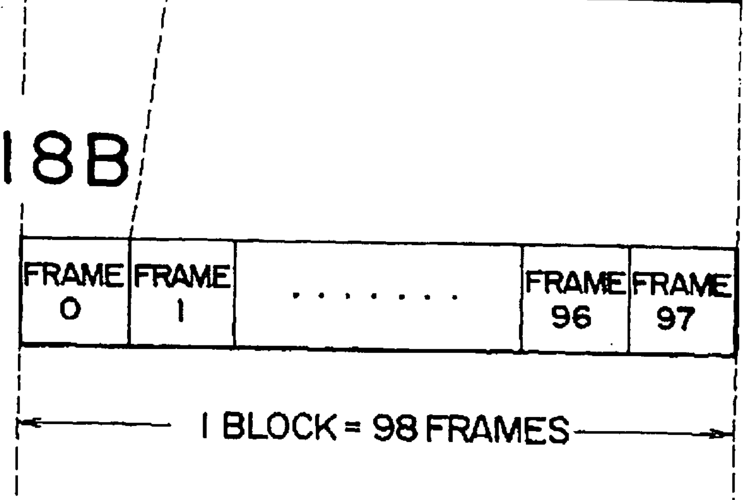
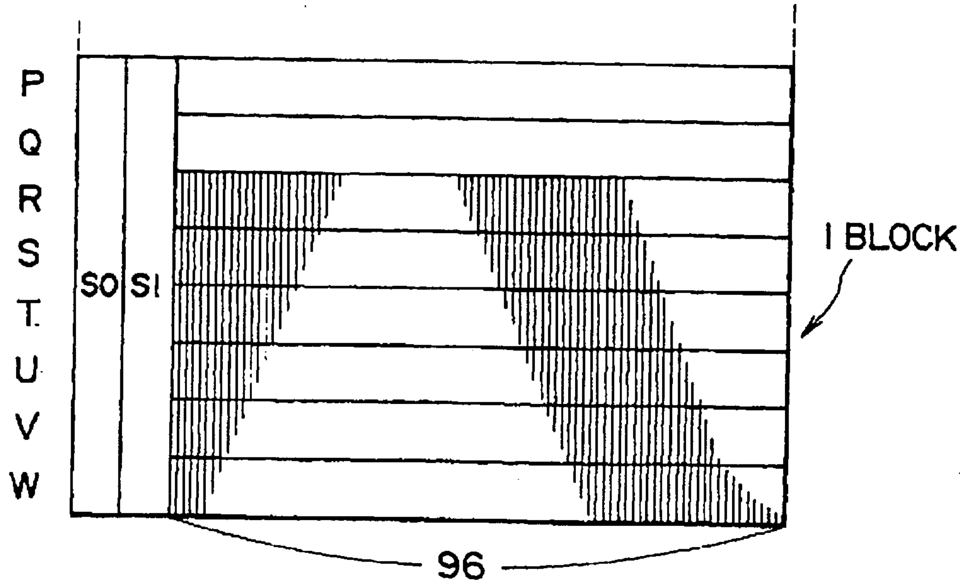


FIG. 18C

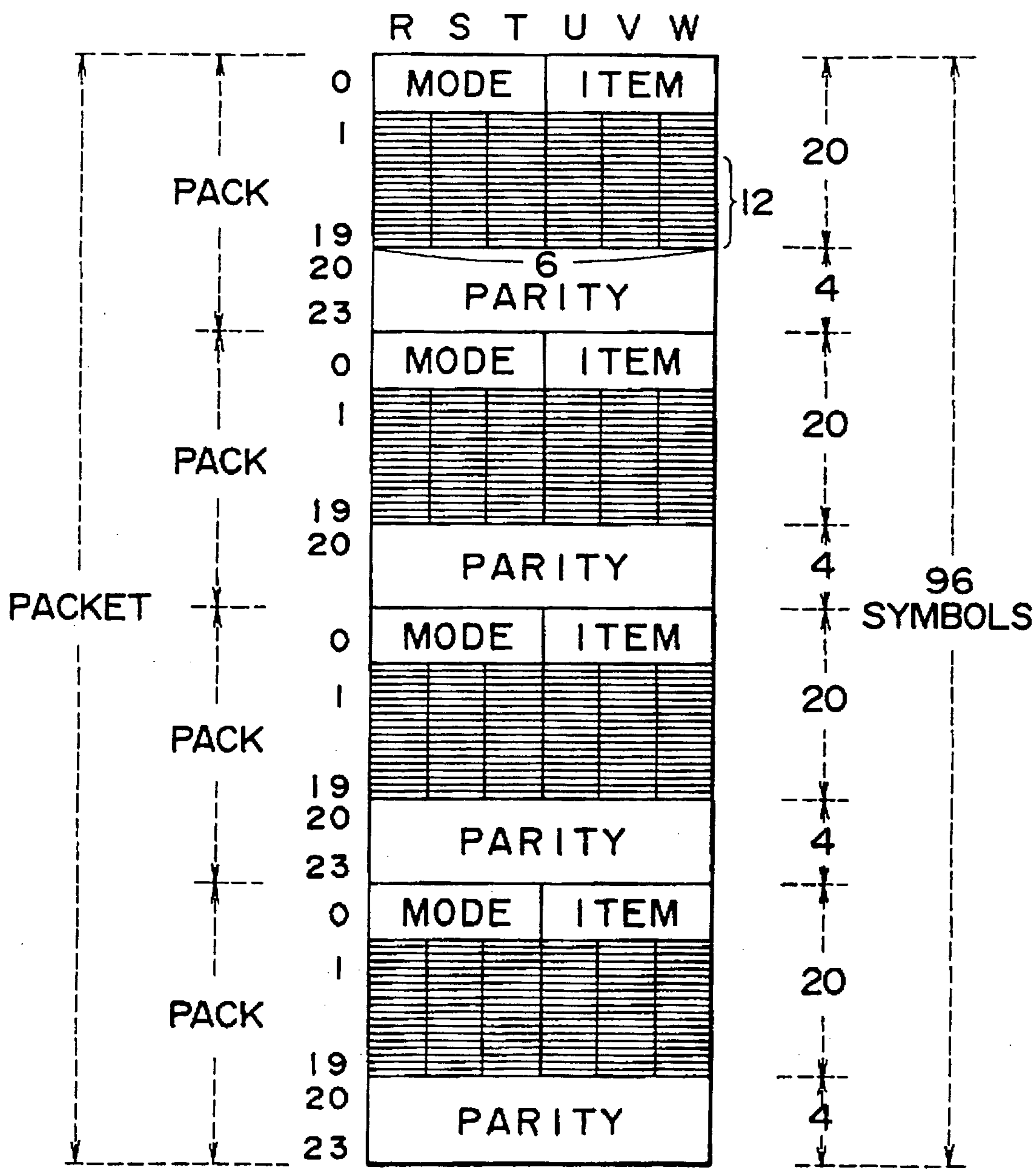


1 BLOCK → 75Hz

1 FRAME → 75 x 98Hz

SUBCODE BIT RATE = 7.35 KBYTES/s

FIG. 19



TRANSMISSION FORMAT